SEMI-Automatic approach to domain ontology building

Henrihs Gorskis, Tatjana Zmanovska, Jurijs Chizhov

Department of Modelling and Simulation, Riga Technical University, 1 Mezha street, Building 4, Room 456, Riga, LV-1048, Latvia
Henrihs.Gorskis@rtu.lv

Abstract: *This paper presents an automated method for building task ontology models from guideline models. A guideline is a specification of steps that need to be taken in certain situations and criteria that need to be fulfilled for these steps to be chosen. The ontology building process concentrates on machine readable guideline models, in particular, on the Guideline Interchange Format (GLIF). Since guidelines for similar processes most likely have many common concepts, it can be proposed that an ontological model of the task domain could be used for information storage and rule extraction.*
In order to accomplish the set goal of building a task ontology model from guidelines, it is necessary to do the following: create or convert basic concepts of the task into the concepts of ontology; create a relational structure between concepts within the ontology, capable of representing the choices and the sequence of the guidelines and unify equal concepts from different guidelines.
The extraction of concepts can be done by finding data request and task execution blocks in the guideline model. Data request blocks would correspond to environmental or system state concepts. To create a class hierarchy of the tasks and concepts for the ontology as well as the relational structure, a thorough analysis of the guidelines is required. This method creates a custom ontology structure and creates new relations that would be able to describe the processes in the guidelines in such a way that rule extraction is possible without keeping the original structure among guideline concepts.
Automated ontology building from guideline models seems to be very realistic and has the potential of combining the practical data from guidelines with the capabilities of ontology models, simultaneously uniting several guidelines into one large shared structure.

Keywords: Domain ontology building, clinical practice guideline.

Introduction

Ontology models are widely used in many scientific fields. Among many they are used in knowledge engineering, artificial intelligence, knowledge management, natural language processing, e-commerce, intelligent information integration, bio-informatics, education, semantic web etc. (Gorskis and Chizhov, 2012). The term ontology in computer science should not be confused with the philosophical meaning used by Plato and Aristotle, describing the nature of being. Building an ontology model is complex work and demands a lot of time. In order to build the ontology, usually a domain expert is required. The task of the expert is to declare all domain concepts and the relationships between them. The ontology is build for a certain field of interest and describes its domain by being a model representation of it. A computer readable ontology provides information for users and software agents about the domain. It can also be used to gain additional information about the domain by analyzing existing relationships between the given concepts.
A guideline is a document that dictates or helps determine a course of action by describing a situation and recommending the best action for that situation. To determine a situation, usually a set of criteria is given that need to be tested. There exist many computer readable document models and languages for the construction of guidelines. This paper proposes the use of these models and the guidelines themselves for the improvement and simplification, as well as acceleration of automated ontology building, by using them as prior information.
This paper takes a look at how computer understandable guidelines (and guidelines in general) can be used as the prior information and foundation of the ontology building process.

Guideline element models

There are several already developed guideline element models that are used to create guidelines. Most guidelines are used in the field of medicine and healthcare. Clinical guidelines are potential tools for standardizing patient care to improve its quality and cost effectiveness (Peleg et al., 2000). Structured, computer-interpretable guidelines can be delivered to the point of care in a way that enables decision support. This makes them very suitable for ontology building. Some of the most used guideline element models are GLIF, GLIF3 and GEM. The GLIF specification is aimed to provide a very precise representation of the guideline. It is designed to be both readable and computable. It is computable in the sense that the logic and sequence in guidelines specified in GLIF can not only be read, but also interpreted by computer. The very first version of GLIF was described in a pseudo code specifically meant to store values that describe the order of execution and related data. This high level code was almost ready for execution on a computer. Since the GLIF has moved on to a model that uses a

UML like language for structuring the guideline. This provides more flexibility, but has a far less clear data structure.
The GEM element model created by the SAGE project has defined an overall methodology for developing medical guideline knowledge bases (Shiffman et al., 2000). It is meant to provide decision support using these knowledge bases.
The GEM element model is meant to be used for guidelines specified in XML. It therefore can be depicted as a directed graph with the "Guideline Document" element as the root. The major concepts in the first tier of the GEM hierarchy below the root level are identity, developer, purpose, intended audience, method of development, target population, knowledge components, testing, and review plan. Each of these elements, in turn, comprises one or more additional levels of guideline constructs. Components of GEM are defined as XML elements. Elements have distinct names and are delimited with start and end tags.

Ontology engineering

An ontology is a specification of a conceptualization. That is, it is a description of concepts and relationships that exists for an agent or a community of agents (Gorskis and Chizhov, 2012). One can also say that the ontology is a formal, explicit description of concepts in a domain of discourse. It consists of classes sometimes called concepts; properties of each concept describing various features and attributes of the concept called slots, roles or properties; and restrictions on slots called facets or called role restrictions. The use, structure and functions of ontology models can be very different; however, the elements contained in ontology models are mostly the same. Among the many ontology definitions available, a clear description is in the following definition: an ontology structure is a 7-tuple O (1)

$$O = \{C, R, H^C, rel, I_C, I_R, A^O\} \quad (1)$$

that contains

- two disjoint sets C and R whose elements are called concepts and relations, respectively;
- a concept hierarchy H^C where H^C represents the hierarchy of concepts in their relations as $H^C \subseteq C \times C$, where H(C1, C2) means that C1 is a sub concept of C2. This hierarchy between concepts can also be called taxonomy;
- a function relation rel: R !C ×C, that relates concepts non-hierarchically. This can be written rel(R)=(C1,C2) or R(C1,C2) Relations that are not within the hierarchical structure and describe a relation between any 2 concepts or 2 individuals. These relations also include attributes;
- the sets I_C and I_R describe instances of the existing concepts (C) and relations (R);
- a set of ontology axioms A^O (additional rules and restrictions), expressed in an appropriate logical language.

From this it becomes clear that an ontology model holds concepts or classes that describe object or phenomena within a domain. The relations between the concepts inform about the hierarchical structure of classes and subclasses and also of other kinds of relations between them. This gives a description not only about the concepts and their place within the model and relative to other concepts, but also a description of the domain itself.
In order to represent ontology models, the use of a standardized language is needed. The Web Ontology Language (OWL) offers the necessary building blocks for ontology representation. The result of an automated or semi automated ontology building process can be an OWL file that holds the structured data of the ontology. OWL extends the Resource Description Framework (RDF) and is based on XML. That means that the ontology is stored as a text file in which the data are structured with the help of tags.

Related works

Even though the ontology building process described in this paper is using preexisting machine readable data, this work concerns all forms of ontology building. This process relies more on the analysis of the given data execution path and does not require data mining approaches, however it could be possible that in future such a necessity could arise.
By using guidelines as preexisting data for ontology building, the result of the process is a guideline stored within an ontology model. It conceptualizes the elements of a guideline, their properties, and defines the relationships that are held among them. For example, all medical guideline representation ontology models have a set of medical decisions and relevant actions (concepts), and a set of temporal rules that relate decision evaluation results to associated actions (relationships). A well established and generally acknowledged guideline representation ontology ensures that the resulting representations can be easily understood by non-authoring human readers, therefore facilitates the dissemination of guidelines across institutions. Well defined computational ontologies also provide considerable promise of enabling automated guideline acquisition,

visualization, execution, and sharing. Such characteristics are prerequisites for a computer-recognizable, interchangeable guideline format. Without these features, it is difficult to enable automated knowledge acquisition and execution for Clinical Decision Support Systems designed to enhance evidence-based practice (Zielstorff, 1998).

Only in the case where an element model for a given guideline is missing and the structure of the guideline itself is not very clear, a data mining approach could be necessary. In the paper by B. Fortuna the ontology building process uses information given by the classification system as hints for how to structure the ontology (Fortuna et al., 2006). The classification is based on SVM with respect to concepts of the domain. For such task the OntoGen software can be used. It is a semi-automatic and data-driven ontology editor focusing on the editing of topic ontologies. The system combines the text mining techniques with an efficient user interface to bridge the gap between the complex ontology editing tools and the domain experts who are constructing the ontology.

E. Blomqvist uses clues given by the concepts of a subject domain (Blomqvist, 2007). They are derived from the analysis of the subject areas with the help of case-based reasoning. Therefore, this approach is also semi automatic.

Suggested approach

The main idea of this paper is to use pre-existing and accessible guide element models as prior information in the ontology construction process. Using the predefined models that describe guideline elements for the basis of ontology building, it is possible to build an ontology model of the information given in the guideline, based on the underlying element structure.

First, it is necessary to obtain the guideline element model used for the creation of the guideline. There are different approaches for this. The simplest way is to obtain the original specification. This is easy if the guideline was written with the GEM or GLIF specification. The use of a predefined guideline element model can lead to excessive concepts in the final ontology in cases where a closely defined ontology would be desired.

When the predefined guideline element model is not given, it is required to create ontology concepts from the elements given in the guideline itself.

Using this approach the main structure of the ontology will be dictated by the element model. First, the main concept is defined. This concept can be a “thing”, or a general ontology concept. All other ontology concepts are related to this main concept. The other concepts are taken from the element model. Elements are translated into ontology concepts and the hierarchical structure of these concepts mimic the structure of the elements from the guideline element model. This first part has simplified the process of ontology building a lot, by virtue of being a translation and copying process.

Depending on the language of the guideline element model it is necessary to create transformation descriptions. For example, GEM is given as an XML schema. By using the information in the GEM file and looking for the element tag it is possible to extract the main concepts and relations. It is also possible to extract concept properties by looking for other information inside the element tag. The extraction of properties and the formats of properties can be more difficult than the extraction of elements alone.

This step needs to be supervised by an expert in order to make sure that only relevant information is being extracted from the model. The expert also needs to make sure that the relations between concepts are in order. In case of an ill defined model, the expert needs to create the necessary relations and add or remove information from the transformation.

The final step of ontology building from guidelines is the creation of concepts instances. For this it is required to be informed of all concepts within the guideline from the element model. By going through the guideline and reviewing the element found in it, instances of the related concept are created in the ontology and filled with the information provided in the guideline description.

Tasks to be solved

In order to extract the basic foundation of the ontology from GEM or any other specification a computerized solution must be created that transforms the GEM or other schema file into the ontology basis as an OWL file. A transformation needs to be created that finds <xs:element> tags and created <owl:class> tags that are given the same name as the “name” property in the guideline element. Further a solution for annotation and documentation needs to be created since it is unclear how there elements need to be treated in the ontology. In the case of GEM tags such as “<xs: complexType>” or “<xs:sequence>” can be overlooked since they are schema specific, but do not add information to the guideline model. All elements found within other elements can be labeled as subclasses. Since instances of elements in the guideline differ slightly from the GEM element model some slight changes can be required. The main difference is that connections between guideline element instances are not defined in the GEM element model. GLIF elements are defined in such a way that connections between elements are expected and regulated. A solution needs to be found for all cases. For a better use of the capabilities provided by the ontology it would be important to create conceptual relations for the guideline concepts as given in the guideline. For that, instances of concepts in the ontology can be viewed as concepts themselves. Relations between such concepts would be defined by the order of steps given in the guideline.

Since the order of steps in the guideline is defined by connections between actions and conditions, the information for rule creation can be extracted by following the execution of the guideline. Having extracted all concepts it is possible to detect what concepts are connected to the conditional actions and create firm or generalized rules based on those concepts. There rules are stored separately from the ontology in the IF/THEN syntax.
Since guideline element models can be very different both in their structure and the way the elements are described, it may be impossible to completely automate the extraction process. Automation at this time may be only capable to extract the general structure of the element model. User input can clarify the exact structure and meaning of the elements in the model.

Implementation of the suggested idea

In order to show the approach of ontology building from a guideline, a small test guideline is used. Fig. 1 shows a simplified guideline element model that will be used in this paper, which describes the overarching classes of information types and the elements that need to be filled with information for a guideline instance.
The first step of the ontology building process is to determine the guideline element model, since the base of the ontology will be based on it.

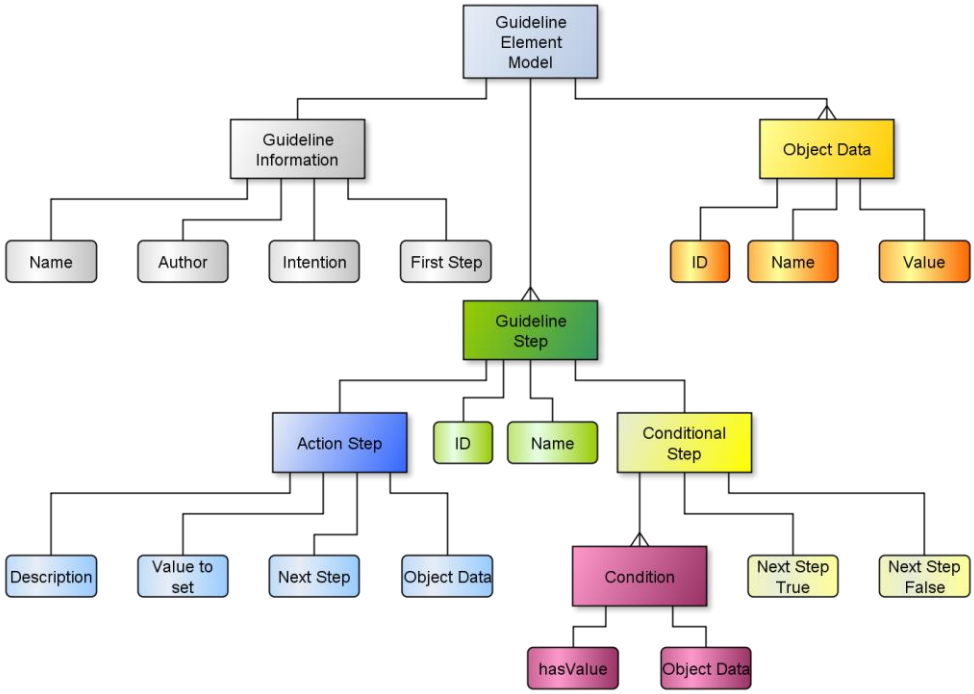

Fig. 1. **Test-guideline element model.**

This simplified element model was created based on a simplification of GEM, hoverer elements like “First step” and “Next step” are taken from GLIF. Every leaf element in this tree describes a certain value that has to be given. The element “Name” is part of the Guideline information, where Guideline information is a general class that does not have any separate information besides its leaf elements. It is important to note that the “name” element from the guidelines information is not the same as the “name” element from the guideline step or object data classes. The “next step” type elements hold information, which is required for the execution of a guideline written with this model. By analyzing the base structure of a guideline written using this element model or by analyzing the guideline itself, it is possible to recognize the root element, classes and subclasses, and inheritance between classes. Right away the ontology building process can start creating ontology classes by copying this information and adding the required notations. For example, the element <guideline.information> would be transformed to the OWL code in Fig. 2.

```
<owl:Class rdf:ID=" guideline.information ">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="guideline.element.model"/>
  </rdfs:subClassOf>
</owl:Class>
```

Fig. 2. **The OWL code example.**

The next step involves the guideline itself. Fig.3 shows a simple guideline about the weather. This is a simple sample guideline about the weather. It will be used to expand on the base structure of the ontology and provide the instance information for the existing concepts.

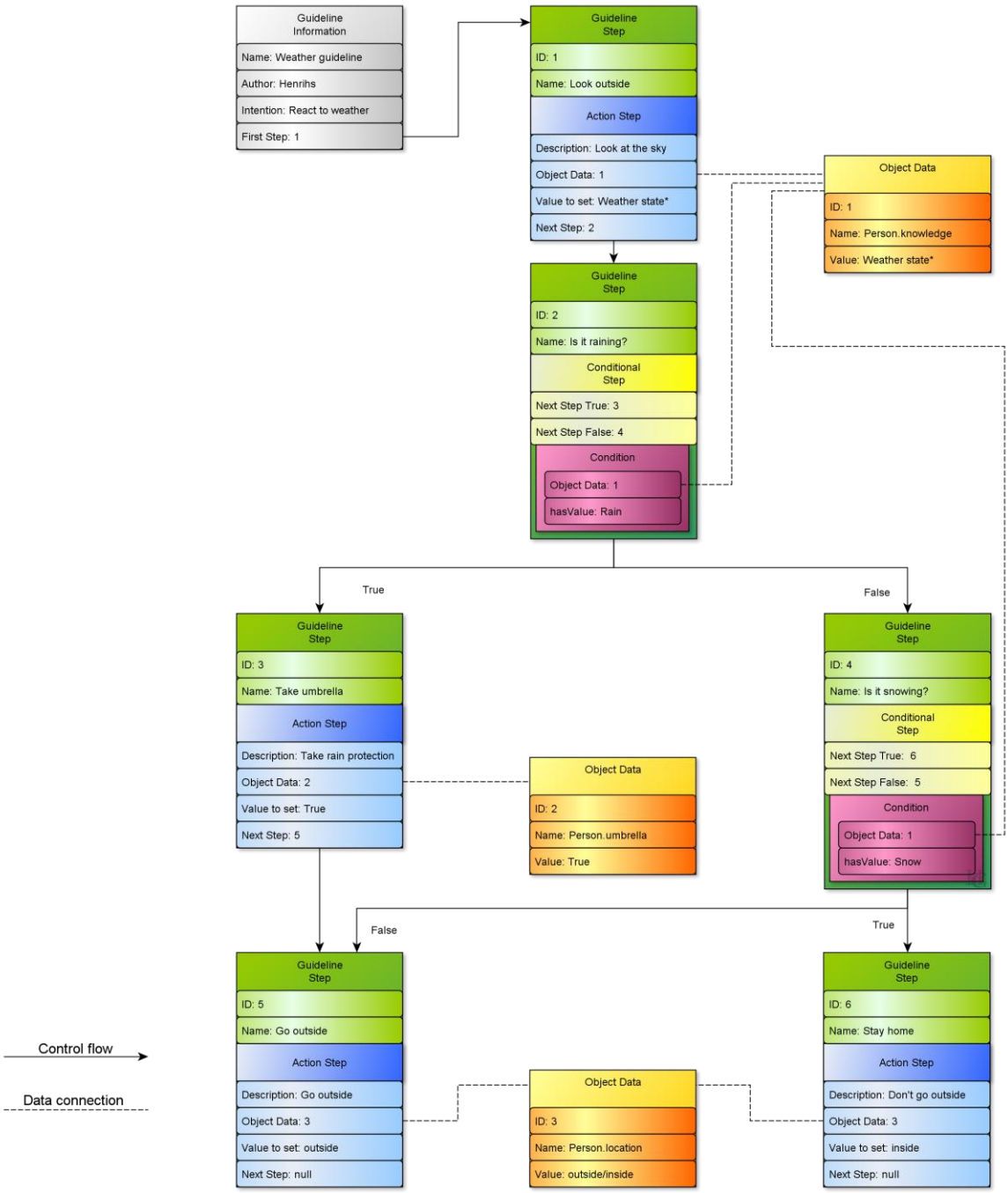

Fig. 3. **A test guideline.**

As one can see, the guideline shown in Fig. 3 consists of elements from the guideline element model. There is only one instance of the "guideline information" element and several instances of the "action step", "conditional step" and "object data" elements. The xml code of this guideline follows the same structure of the guideline element model. Any elements that are used more than one time are written separately and are filled with the information that is shown in Fig. 3. Since the "action step" and "conditional step" elements both inherit from the element "guideline step", the xml code for these elements will begin with the mention of the parent "guideline step" elements and hold information for "ID" and "name", however the body of the "guideline step" will only feature an "action step" or an "conditional step" element, but not both.

The third step takes the information from the guideline and adds instance data to the ontology that is based on the element model as is shown in Fig. 4.

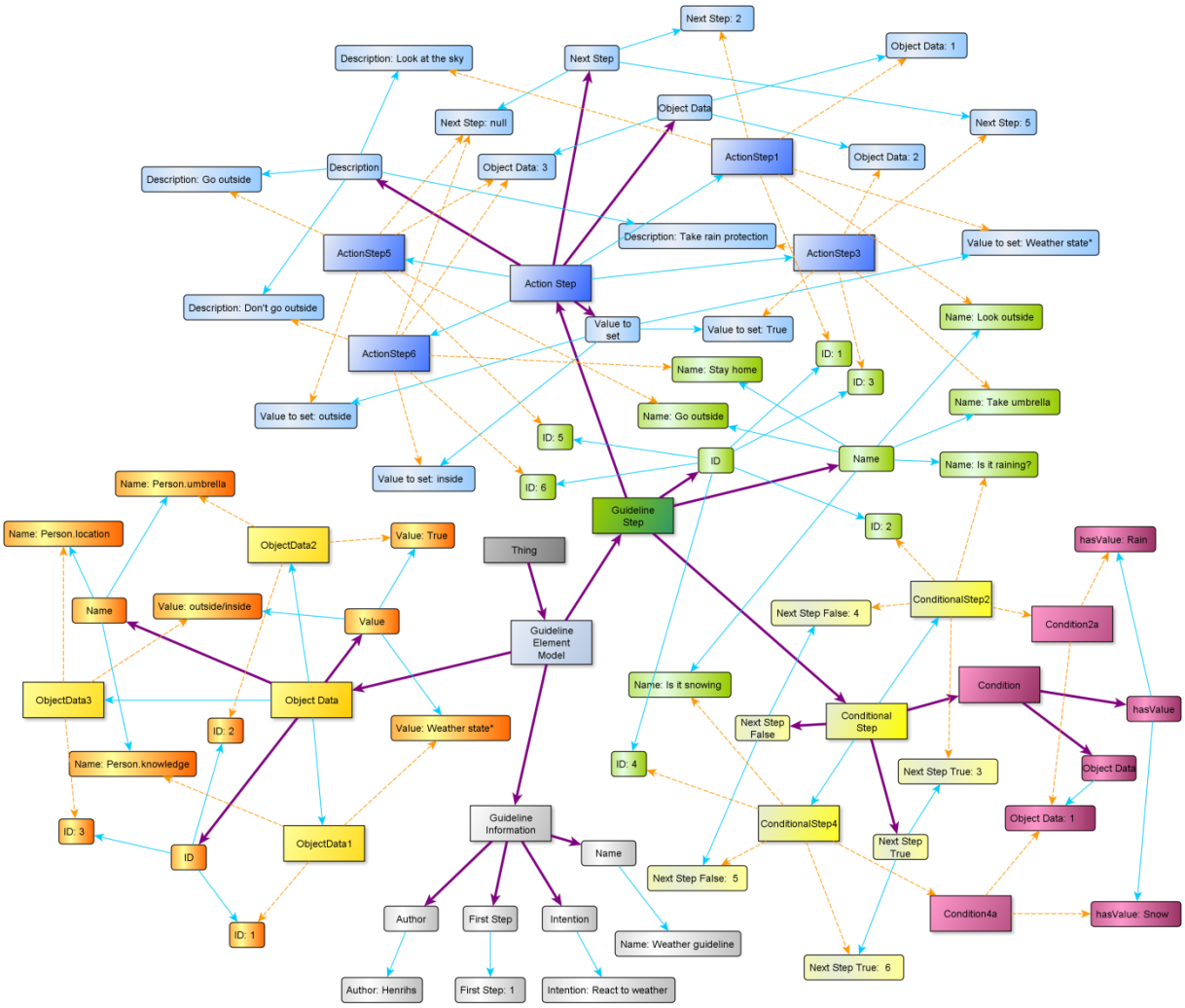

Fig. 4. **Test ontology from guideline data.**

All the information of the guideline has been implemented into ontology as instances of the data elements they belonged to. It should be noted that any classes that are similar in name, but are subclasses of different parent classes are not the same class. Any instance class that has the same information as an instance class that was already added is not includes, so that there is only one instance with the unique data. In order to maintain the execution information and the data described by the guideline additional classes have been created to store the relations between instances of data classes and instances of the overarching unifying classes. These are classes like "ActionStep1", "ConditionalStep4" and "ObjectData3". These are generated classes that are meant to store the relations between the abstract parent class and the instance of these classes in the guideline. These instances of the abstract parent classes are required to fulfill the idea of connecting instances only with other instances.

As the ontology is built, rule extraction operations can be performed in order to obtain additional rules about the guideline and the routes given in the mentioned guideline. As the guidelines data are analyzed, IF/THEN rules can be extracted about what data relate to each other in the context of the guidelines algorithm.

The resulting ontology should provide all the data that were given by the guideline and the element model it was based upon. This would conclude this method of ontology building.

Conclusion

This paper proposed a method of ontology construction by using information from and about a computer readable guideline. The obtained ontology summarizes the guideline elements and the information provided in the guideline itself. Even though ontology models like that could be usable right away, it is recommended and often even required to do additional work on the ontology by an expert.

The proposed method can be easily extended to use many guidelines of the same type for the acquisition of prior information. Any additional guideline would add more instance classes of the base classes. However in order to distinguish between information from different guidelines, additional information needs to be included in the ontology for the purpose of maintaining separation between data. One way of doing this is to add an instance of a "guideline" element that would be related to those and only those instances that were given in the separate guideline.

Future work can include the design and development of a computerized tool, which performs the described steps and generated the ontology model as OWL code.

Acknowledgements

Thanks to Dr.habil.sc.comp. Professor Arkady Borisov (Riga Technical University) for help and support.

References

Blomqvist, E., 2007. Semi-automatic Ontology Engineering using Patterns, ISWC/ASWC 2007 pp. 911-915.

Fortuna, B., Grobelnik, M., Mladenić, D., 2006. Semi-automatic data-driven ontology construction system, PASCAL EPRINTS – WORKING GROUP SUMMARY 15

Gorskis, H., Chizhov, Y., (2012). Ontology building using data mining techniques. Scientific Journal of Riga Technical University, Information Technology and Management Science, Vol.15, pp. 183-188.

Peleg, M., Boxwala, AA., Ogunyemi, O., 2000, GLIF3: the evolution of a guideline representation format. Proc AMIA Symp, pp. 645-649.

Shiffman, R., Karras, B., Agrawal, A., 2000. GEM: a proposal for a more comprehensive guideline document model using XML, J Am Med Inform Assoc., 7(5) pp. 488-98.

Syed Sibte Raza Abidi, Shapoor Shayegani, 2009. Modeling the Form and Function of Clinical Practice Guidelines: An Ontological Model to Computerize Clinical Practice Guidelines Knowledge Management for Health Care Procedures, Lecture Notes in Computer Science Volume 5626, pp. 81-91.

Zielstorff, R., 1998. Online practice guidelines: issues, obstacles, and future prospects, Journal of the American Medical Informatics Association, 5, pp. 227-236.