Medical Informatics Europe 2002
G Surján et al. (eds)
IOS Press 2002

Evaluation of a frame-based Ontology A formalization-oriented Approach

Ronald CORNET, Ameen ABU-HANNA

Academic Medical Center, dept. of Med. Informatics,
P.O.Box 22700, 1100 DE Amsterdam

Abstract. This paper describes the evaluation of our current frame-based ontology in Intensive Care. Our approach to the evaluation consists of formalizing the ontology. The motivation for formalization is twofold: a. The formalization process itself can shed light on the implicit ambiguities in the ontology and b. The result of this process is a representation that facilitates automatic inference. The evaluation has revealed various ambiguities in the current ontology, and has clarified the path for migration towards a formal representation supporting automatic inference.

1. Introduction

Since the last decennia, a shift has been taking place from hierarchical representation of terminological systems to (more) formal representations with support of semantic links between concepts. The semantic web is a well-known example, reflecting this shift of representation in web-documents. In medicine, GALEN [1] and SNOMED [2] are two important examples of moving toward formal, logic-based terminologies.

Our department is engaged in a continuous effort to develop a Terminological System (TS) and corresponding software for the domain of Intensive Care (IC). This TS, called DICE, comprises reasons for admission to IC including diagnoses and surgical procedures [3]. DICE can be characterized as having an explicit conceptual frame-based specification and provides a multilingual thesaurus that supports synonymy. In addition, the specification is augmented by logical constraints, which are however not enforced in the implementation. Although the frame-based formalism has provided a useful starting point for the construction of the ontology, there are a number of limitations of the formalism that lead to problems in practical use: the ambiguous knowledge representation hampers automatic inference, which prevents the continuous maintenance of the terminological knowledge.

This paper describes our efforts to move from a frame-based ontology to a formal representation. This shift implies scrutinizing the ontology in terms of ambiguities and a move from reliance on ad-hoc programs that interpret the ontology to automatic and principled inference with a formal ontology. We sketch a path for migration from our current frame-based ontology to one based on Description Logic and show the problems encountered and advantages gained by this migration. We compare the resulting Description Logic to the ones used in GALEN and SNOMED.

2. Design of DICE

DICE (Diagnoses for Intensive Care Evaluation) is an architecture for terminology-based patient data recording (see [3] for an elaborate description). It consists of a Terminological System, a Terminology Server, and a Terminology Browser. A *Terminology Server* provides functionality through an Application Programming Interface (API) for navigation through and manipulation of the Terminological System. This server is accessible through a Graphical User Interface in the *Terminology Browser*. The *Terminological System* represents the actual domain ontology: knowledge interrelating concepts, in this case diseases and admission reasons to the Intensive Care (e.g. by specifying that viral hepatitis is a special kind of infective hepatitis).

The current DICE ontology is frame-based: classes represent concepts, and slots represent binary relations between concepts. Table 1 shows some examples of class specifications. Slots and slot-values are inherited from superclass(es), but can be overridden. In DICE, two slot facets have been added to the regular frames formalism: First, the explicit labeling of part-of slots because of their importance in medical knowledge: these transitive slots form the basis for constructing a mereology (a system of concepts related by their part-whole relations). Second, the possibility of labeling a slot as a qualifier for a class. For example, the qualifier tag "one_or_more", in the last row in Table 1, indicates that classes can be composed (i.e. post-coordinated), and hence not all possible combinations of slot-values have to be specified beforehand, as this could lead to combinatorial explosion.

Table 1: Examples of frame-based class specifications

Class	**SuperClass**	**Slot**	**Slot-Value**
Hepatitis	Health Problem	Involved Tractus	Digestive System
		Localization	Liver
Infective Hepatitis	Hepatitis	Abnormality	Infection
Viral Hepatitis	Infective Hepatitis	(One_or_more) Etiology	{ Hepatitis Viruses, Cytomegalo Virus, Epstein Barr Virus }

3. Evaluation

The formalism of the ontology in the Terminological System determines the limits of the functionalities that a Terminology Server can provide: the measure of support for automated classification and consistency maintenance. We focus on the evaluation and formalization of the ontology because the formalization process itself should shed light on the implicit ambiguities in the ontology, and the result of this process is a representation that can support automatic inference.

The current Terminological System, based on our frame-based formalism, consists of approximately 2400 classes and 45 slots. It has proven to be a useful formalism because of the possibility to attach a class to multiple superclasses (parents), and to search for classes based on (simple) criteria.

The major drawback of a frame-based formalism is the lack of explicit semantics [4]. This is generally not a problem for a human reader, being capable of interpreting specifications as shown in Table 1. However, continuous quality assurance of the terminological knowledge requires automatic interpretation of and inferencing with knowledge. This is hampered by three important types of ambiguities of frame-based systems that we experienced in practice. We will use the examples from Table 1 to illustrate these ambiguities.

Complete or incomplete specifications. The specification for hepatitis in Table 1 is not complete. This means that the specified slot-values contain necessary, but not sufficient conditions to define hepatitis. Hence, not every Health Problem in the liver and involving the digestive system is hepatitis. The specification for Infective Hepatitis is considered complete. Anything that is specified as hepatitis with an infection **is** an infective hepatitis. This means that every class being specified as such could be automatically classified as infective hepatitis. Not making explicit whether a specification is complete or incomplete allows improper placement of a class in the hierarchy, e.g. during the knowledge acquisition process, and requires more human intervention from a modeler when adding or removing a class.

Distinction in semantics of slots. Three possible interpretations of a slot-value are: restriction, possible value, and necessary value. The slot-value pair "localization-liver" in the first example of Table 1 could mean: Hepatitis can be located in the liver (but does not have to be); Hepatitis must be located in the liver (but can additionally be located elsewhere); Hepatitis is always and only located in the liver.

Depending on the assumed meaning, "hepatitis located in kidneys only", or "hepatitis also located in kidneys" can be a valid specification. Hence, a modeler gets little support in consistent specification of classes, and querying (e.g. "can hepatitis be located in kidneys?") is limited.

Explicit semantics of multi-valued slots. Multi-valued slots, for example the value "(HepA virus, HepB virus, HepC virus)" for the "has_etiology" slot, can have the following interpretations: only one, one or more, or all of the values are valid. This is a frequently occurring special case of the ambiguity in semantics of slots mentioned above.

Due to these ambiguities in the semantics, it is impossible to automatically place classes in the class-hierarchy; detect redundancies in class specifications; detect inconsistencies in class specifications; and query for classes meeting certain complex criteria.

Again, these do not only form a theoretical problem, but they have real drawbacks in practice. Although the current ontology is not extremely large (about 2400 classes, and 45 slots), a number of misplaced, duplicate or redundantly specified classes were already found in the first steps of formalization. For example: CABG is correctly specified as a subclass of both "bypass/graft of artery" and "coronary artery procedures", but redundantly specifies two slot-values that it inherits from its superclasses: "localization: coronary arteries" (inherited from "coronary artery procedures"), and "act involved: bypass/graft" (inherited from "bypass/graft of artery"). It is expected that more inconsistencies and redundancies will be found during the further formalization process, especially by stating disjointness of concepts. However, even if this is not the case, the formalization leads to a trustworthy ontology, because of the thorough examination.

4. Migration

To overcome the ambiguities in the specification of a class, as described in the previous paragraph, a semantically more explicit formalism is required. Description Logics (DLs) can fulfill this requirement. Table 2 shows correspondence between Frame and DL terminology. Table 3 shows options for possible translations of frame specification examples to DL statements. Migrating a frame-based ontology to a DL-based ontology can be done in a number of ways, which we will discuss regarding their implications on the three types of ambiguities. We have used RACER [5], a DL reasoner, to evaluate the various versions of DL-based ontologies with regard to their consistency. The choice for RACER was based on the expressiveness of the DL it supports, its availability, and its comprehensive set of supported queries.

Table 2: Correspondence between Frames and DL terminology

Frames	Description Logic
Class	Concept
Slot	Role
Slot-value	Role-filler
Subclass	Child , subsumee
Superclass	Parent , subsumer
Incomplete specification	Primitive Definition
Complete specification	Definition

4.1. Definitions

Interpreting all frame-based class specifications as primitive concept definitions in DL results in a consistent ontology but hampers automatic classification of concepts. However, considering all definitions complete will lead to concepts that are regarded equivalent in the ontology, as they will have the same definition. Consequently, those classes solely defined as a subclass of exactly one other class (for example: “hepatitis A virus” IsA “hepatitis virus”) should be regarded as primitive, otherwise they would express equivalent concepts. During our evaluation, we have defined all other classes (i.e. those classes not solely defined as a subclass of exactly one other class) as non-primitive, as this facilitates detection of concepts with equivalent definitions, which might indicate existence of duplicate concepts. Furthermore, the equivalencies can be used to determine the usefulness of extending concept definitions by the introduction of new roles. 1351 concepts were defined as non-primitive (the other concepts that were defined as primitive involved mainly etiologies and anatomical components). Classifying the resulting ontology with RACER showed that 857 concepts where defined uniquely, and 494 concepts were non-uniquely defined, hence these had one or more equivalent concepts. The number of equivalently defined concepts ranged from 95 pairs (many of which contained “opposite” concepts such as “hypotension” and “hypertension”) up to one group of 16 concepts, all defined as “direct health problem located in body fluids, involving the metabolic system”. This evaluation was repeated with various interpretations of the semantics of slots and slot-values, as described below, but this was not of influence, as in each step all slots were interpreted similarly.

The primitive concepts should be carefully reviewed in order to determine mutually exclusive concepts. Note that stating disjointness, (e.g. “disjoint {virus, bacterium}”, to specify that nothing is both a virus and a bacterium), has no equivalence in frame-based logic (apart from the unique name assumption that everything named differently is different), but plays an important role in restricting role-fillers, as explained below.

Table 3: Translations of frame-based class specifications to Description Logic-based concept definitions

Frames	Description Logic
ViralHepatitis: Hepatitis, ViralDisease	(Defprimitiveconcept ViralHepatitis (and Hepatitis ViralDisease)) (Defconcept ViralHepatitis (and Hepatitis ViralDisease))
ViralDisease: Disease hasCause Virus	(Def[primitive]concept ViralDisease (some hasCause Virus)) (Def[primitive]concept ViralDisease (all hasCause Virus))
Kidney: haslocation (left, right)	... (some hasLocation (or left right)) ... (some hasLocation (and left right)) ... (and (some hasLocation left) (some hasLocation right))

4.2. Semantics of Slots

As with definitions, there are a number of ways to specify the quantification of roles. The first way is to define all roles as existential (i.e. "some"), which enforces presence of such a role, but does not constrain adding other role-fillers for the same role. SNOMED defines roles in this way [2]. Another way would be to define all roles with universal quantification (i.e. "all"), but this might be too restrictive with regard to the domain of the role-filler. The third possibility is a combination of existential and universal quantification, enforcing that at least one role-filler exists, and that all role-fillers are (children of) the specified concept. The classification of both the existential and the universal variant of the ontology in RACER and combinations thereof did not yet present inconsistencies, as no disjointness axioms had been specified yet.

4.3. Multi-valued Slots

An inventory of multi-valued slots shows that they are used with various semantics mentioned in the previous paragraph, hence manual review is required to determine which semantics are intended. Although this is a labor and knowledge intensive effort, it highly increases the functionality for advanced querying of the terminological knowledge.

Having described the ambiguities found in the frame-based formalism, we now discuss the implications of post-coordination and part-whole slots on the resulting DL.

4.4. Post-coordination

It can be debated whether rules for post-coordination should be put in or out of the ontology, as these rules do not address factual knowledge about concepts in the ontology. The rules can be regarded as knowledge about what is (or can be) *known* about concepts, so-called epistemology. Work is being done on inclusion of an epistemic operator in DLs [6], but as it severely hampers the computational properties of the logic, we have currently decided not to include these rules. We foresee that definition of rules outside of the ontology and validation of post-coordinate concepts by means of automatic inference will provide a satisfactory mechanism for supporting post-coordination.

4.5. Part-whole Slots

The explicit specification of part-whole slots in the frame-based formalism can be covered in various ways. The most straightforward way is to define these as transitive roles. However, for more advanced reasoning, one might consider adding role chaining to this. This facilitates reasoning about properties of the whole by applying properties of the constituting parts. Another solution is to introduce Structure-Entity-Part (SEP) Triplets as proposed by [7]. As role chaining increases the computational complexity of the logic, we currently choose to define partitive roles as transitive. However, migration to SEP-triplets must be considered in the future.

5. Comparison with GALEN and SNOMED

The DL that we propose to migrate to contains the following concept constructors: intersection ("and"), universal quantification ("all"), existential quantification ("some"), union ("or") and transitive roles. This logic is called $\mathcal{ALC}_{R^+}$ [5]. This DL has proven to be tractable, and is (apart from role chaining) more expressive than GRAIL (the GALEN DL) and $\mathcal{EL}$, that SNOMED uses. This logic hence supports more expressive concept definitions, preserving computational strength, and facilitates advanced querying, from which both modelers and users profit.

6. Conclusion

Although there is still a long way to go to complete the migration process that we have depicted above, it has already proven its value in determining redundancies in concept definitions and in forcing the modeler to be aware of the ambiguities. A formalism that supports expressive, valid and consistent definition of concepts and proper designation of terms and synonyms is essential for the new generation of Terminological Systems and for trusting their contents.

Acknowledgements

This work is supported by the Netherlands' Organization for Scientific Research (NWO)

References

[1] Rector AL, Nowlan WA. The GALEN project. Comput Methods Programs Biomed. 1994; 45(1-2):75-8.

[2] http://www.snomed.org/, last visited 8-5-2002

[3] De Keizer NF, Abu-Hanna A, Cornet R, Zwetsloot-Schonk JH, Stoutenbeek CP. Analysis and design of an ontology for intensive care diagnoses. Methods Inf Med. 1999;38(2):102-12.

[4] Woods, W. "The KL-ONE Family." Computers Math. Applic. (1992); 23: 133-177.

[5] http://kogs-www.informatik.uni-hamburg.de/~race/, last visited 8-5-2002

[6] Donini FM, Lenzerini M, Nardi D, Nutt W, Schaerf A, An epistemic operator for description logics, Artificial Intelligence, 1998; 100(1-2):225-274

[7] Schulz S, Romacker M, Hahn U. Part-whole reasoning in medical ontologies revisited--introducing SEP triplets into classification-based description logics. Proc AMIA Symp. 1998:830-4.