Contribution of the Ontology Engineering to Mathematical Knowledge Management

Frédéric FÜRST, Michel LECLÈRE & Francky TRICHET
IRIN
Computer Science Research Institute
University of Nantes
2, rue de la Houssinière BP 92208
44322 Nantes (France)
{furst,leclere,trichet}@irin.univ-nantes.fr

1 Introduction

Mathematics represents a field of knowledge which is very structured but also extremely vast. Searching information from documents related to this area (for instance, documents available on the Web) can be very difficult. A solution to this problem consists in automating information retrieval by integrating mathematical knowledge in Information Systems. From the field of *Artificial Intelligence* and more precisely *Knowledge Engineering*, these systems are called Knowledge-Based Systems (KBS). Heirs to Expert Systems, KBS are currently used for different purposes such as Enterprise Knowledge Management (capitalization, share and appropriation of know and know-how), Education (exploitation of intelligent tutoring systems for managing a cooperative problem solving process) or « Semantic Web » (indexation and annotation of web resources by expliciting their semantics).

The main problematics underlying the construction of a KBS is *knowledge representation* which consists in formalizing, from informal descriptions usually expressed in natural language, the knowledge characterizing a specific domain (*i.e.*, the concepts, the relationships and the axiomatics which define, in a nonambiguous way, the semantics of the considered domain). This activity lies with recent works on *Ontology Engineering* [GUARINO & al. 2000, BACHIMONT 2001]. This engineering is concerned with the development of methodologies, models, tools and languages dedicated to the construction of ontologies.

The Artificial Intelligence literature contains many definitions of an ontology (many of these contradict one another). A definition, which seems to be a consensus, can be expressed as follows: *an ontology is a formal explicit description of concepts in a domain of discourse (classes sometimes called concepts) and relations among them* [GRUBER 1993]. An ontology can be developed for different goals, for instance to share common understanding of the structure of information among people or software agents, to enable reuse of domain knowledge or to make domain assumptions explicit.

The work introduced in this paper is concerned with the formalization of the ontology underlying the projective geometry. This formalization is done by using the Conceptual Graphs, a formal model defined in the Artificial Intelligence community. Through this experiment, we want to show that applying knowledge representation techniques to mathematical field is a relevant way to improve the reliability and efficiency of tools dedicated to mathematical knowledge management. Our proposal is based on the construction of knowledge bases (defined according to ontologies) which must be considered as the heart of any mathematical knowledge management tool such as specific math search engines on the web, specific math intelligent tutoring systems, specific math theorem provers, etc.

Note that the idea we advocate begins to be applied in the Web area as shown by the recent development of Semantic Web languages dedicated to the sharing of ontologies and knowledge bases such as RDF/RDFS - « Resource Description Framework » recommended by the W3C and technically based on XML [RDFS 2000] - or the more powerful (from the expressivity point of view) DAML+OIL - DARPA Agent Markup language [HENDLER 2001] / Ontology Inference Layer [FENSEL & al. 2000] - which is based on Description Logics, a model close to the Conceptual Graphs model.

In the following, we present the general principles (illustrated with examples) of the formalization of the projective geometry with Conceptual Graphs. We also emphasize how the formal representation we have defined allows expliciting and automating theorem proofs, and query answering.

2 Representing the projective geometry with Conceptuals Graphs

Our work is based on HILBERT's book, *« Grundlagen der Geometrie »* [HILBERT 1997]. In this book, HILBERT presents the axioms on which he builds the geometry. Five groups of axioms are distinguished: the membership's axioms, the order's axioms, the congruence's axioms, the parallelism's axioms and the continuity's axioms. The projective geometry is built on the first and the second groups of axioms. These axioms have been modelized using the Conceptual Graphs model.

The Conceptual Graphs model, first introduced by Sowa [SOWA 1984], is a knowledge representation model which belongs to the semantic networks. This model uses a graphical notation based on the definition of concepts and relationships between concepts. Knowledge represented with Conceptual Graphs (CGs) is managed by way of graph algorithms. Furthermore, CGs own a logical interpretation of its primitives; this allows giving a formal semantics to the knowledge represented with CGs and to the reasoning mechanisms.

Knowledge expressed within this model is structured into three levels:

- a *terminological level*, used to define the conceptual vocabulary of the considered domain;
- an *ontological level*, used to specify the definitions, the rules and the constraints underlying the considered domain;
- an *assertional level*, used to design graphs representing specific assertions of the domain (*e.g.* theorems) or queries of the end-user. These graphs are constructed by using the conceptual vocabulary.

2.1 The terminological level

The terminological level is composed of a poset of concept types T_c (possibly structured as lattice) and a poset of relation types T_r. The relation types represent the different kinds of relationships which can be stated between instances of concept types. To each relation type is associated a signature which specifies its arity and the maximal concept types of its arguments. Note that the relationships between concept types, such as the *Kind-of* link, are not represented with a relation type; they are represented with the two specialisation/generalisation relations $\leq_c$ and $\leq_r$ which organize the hierarchies T_c and T_r.

Figures 1 and 2 present the concept types T_c and the relation types T_r defined for the domain of projective geometry. The basic objects manipulated in projective geometry are points, straight lines and plans.

Note that because the negative expressions do not exist in CGs, we must add to the hierarchy of relation types the negative forms of the membership relation (*e.g.*, `belongsPS(Point,Set of Points)` versus `nBelongsPS(Point,Set of Points)`).

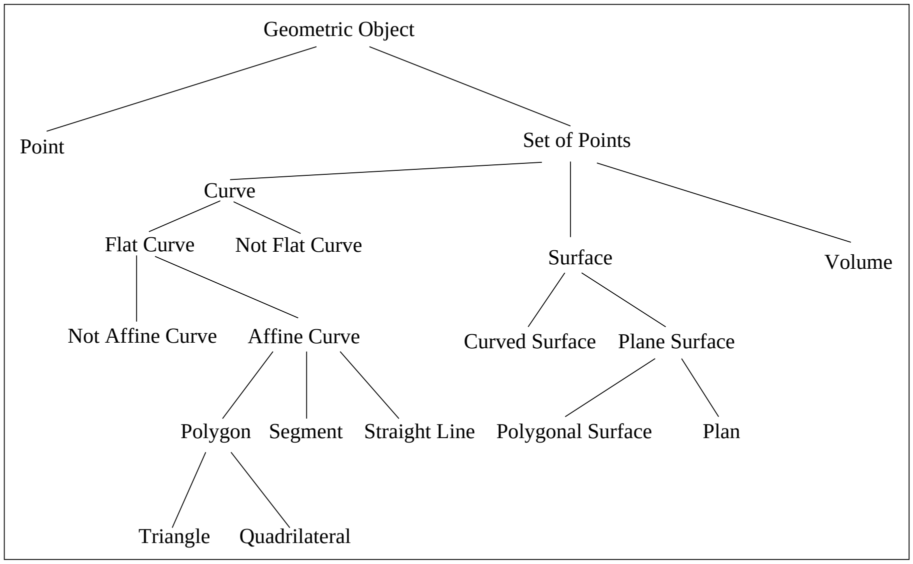

Figure 1: The hierarchy of concept types.

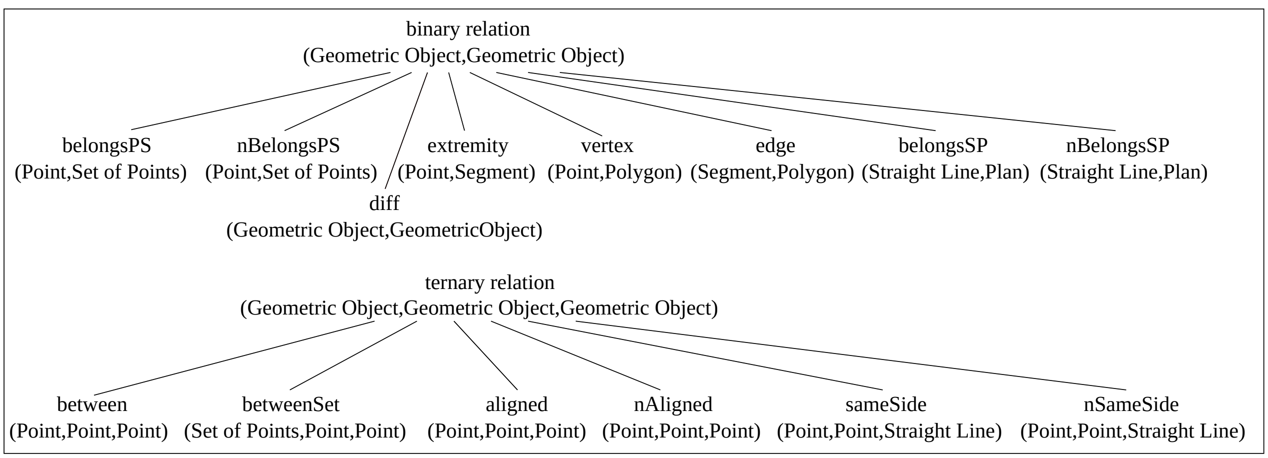

Figure 2: The hierarchy of relation types.

2.2 The ontological level

The ontological level is used to represent the semantics of the considered domain and, in particular, to make the implicit knowledge explicit. Knowledge defined at this level is represented by Conceptual Graph Rules, Conceptual Graph Constraints and Concept/Relation Type Definitions. These primitives are based on the notion of Simple Conceptual Graph (SCG).

A Simple Conceptual Graph (SCG) is a bipartite graph composed of concept vertices (representing objects of the domain) and relation vertices (describing relationships between objects). A specification/generalisation relation $\leq$, based on $\leq_c$ and $\leq_r$, is defined between conceptual graphs. The projection is the fundamental operation on simple graphs since it allows the effective computation of the $\leq$ relation. This operation is essentially a labelled graph homomorphism (cf. section 2.3).

A Conceptual Graph Rule is an inference rule of the form $G_1 \Rightarrow G_2$, where G_1 and G_2 are SCGs (see [SALVAT & al. 1996] for details). The graphical representation of a rule is the same as the representation of a simple graph, except that the hypothesis of the rule is indicated in white color and the information added to the graph by the application of the rule is indicated in black color (cf. figure 4). According to a graph G, a rule is applicable if there exists a projection of the hypothesis graph into G. In this case, the conclusion graph of the rule can be joined to G.

A Conceptual Graph Constraint defines conditions that a simple graph must respect to be considered as valid (see [MUGNIER 2000] for details). It is composed of a condition part and a mandatory part. We consider *positive* and *negative* constraints. A *positive constraint* expresses that « if information A is present, then information B must also be present ». Roughly said, a graph G satisfies a positive constraint if, for every projection of its condition part, there exists a projection from its mandatory part into the graph G. A *negative constraint* expresses that « if information A is present, then information B must be absent ». The graphical representation of a constraint is the same as the one used for Conceptual Graph Rules (cf. figure 3).

A Concept Type Definition asserts an equivalence between a concept type and a monadic abstraction (see [LECLERE 1997] for details). This abstraction represents a set of sufficient and necessary conditions to belong to its type. Any object recognized by the description must belong to the type and any instance of type owns the attributes of the description. We denote $t_c(x) \overset{def}{\Leftrightarrow} D(x)$ the definition of type t_c with x the variable of formal parameter. In the aristotelician approach, the genus of new type is the type of the concept marked by x and $D(x)$ represents the differentiae from t_c to its genus (cf. figure 8).

A Relation Type Definition asserts an equivalence between a relation type and a n-ary abstraction. We denote $t_r(x_1, x_2...x_n) \overset{def}{\Leftrightarrow} D(x_1, x_2...x_n)$ the definition of type t_r with $x_1, x_2...x_n$ the variables of formal parameters (cf. figure 9).

2.2.1 The membership's axioms

HILBERT gives eight axioms which define the membership relation type. All these axioms are represented by using CG rules and/or CG constraints. The second axiom (called Axiom 1-2) is expressed (in natural language) as follows: *There is at most one straight line to which belong two points A and B.* This axiom is represented by the negative constraint of the figure 3.

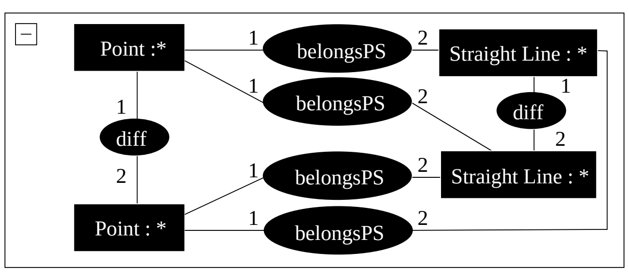

Figure 3: Representation of Axiom 1-2.

The sixth axiom (called Axiom 1-6) is expressed (in natural language) as follows: *If two points A and B which belong to a straight line d also belong to a plan α, then all points of d belong to the plan α.* This axiom is represented by the rule of the figure 4.

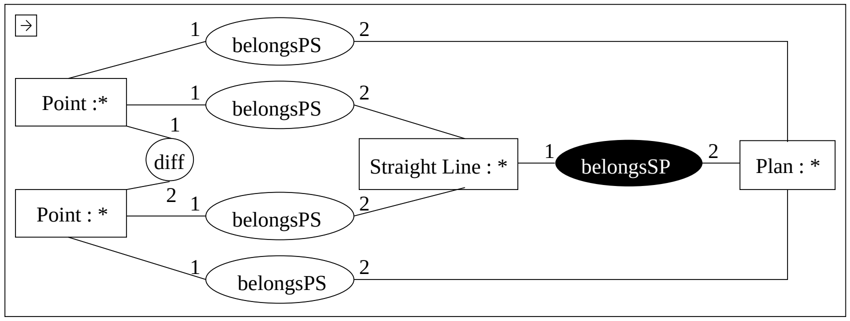

Figure 4: Representation of Axiom 1-6.

Note that some implicit knowledge are not explicitly expressed by HILBERT but must be represented for constructing an operational system. For instance, the figure 5 presents the rule underlying the following situation: *Let A and B be two points, A belongs to a straight line d and B does not belongs to d, then A and B are necessary different points.* In a similar idea, the implicit knowledge underlying the membership relation specifies that it is not possible that an object both `belongs` and `nBelongs` to another object (cf. figure 6).

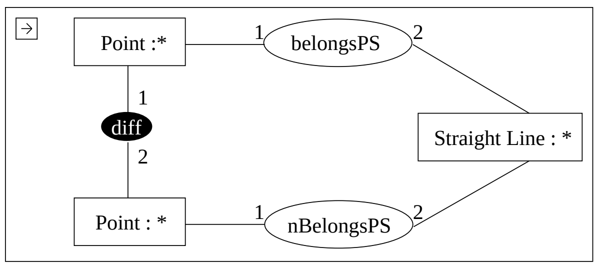

Figure 5: Representation of implicit knowledge.

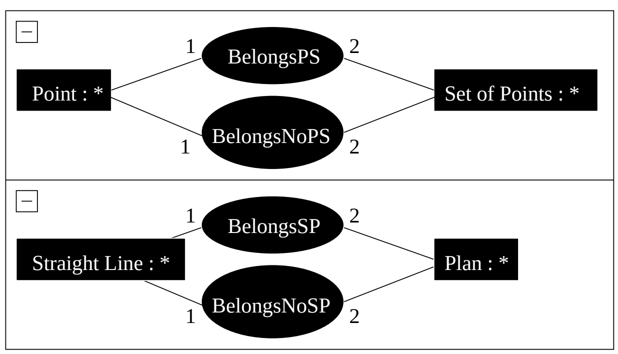

Figure 6: Representation of the membership constraint.

2.2.2 The order's axioms

The four axioms related to this group define the relationship « between ». The first axiom (called Axiom 2-1) is expressed (in natural language) as follows: *If a point B is between a point A and a point C, the three points belong to the same straight line and B is between C and A.* This axiom is represented by the rules of the figure 7.

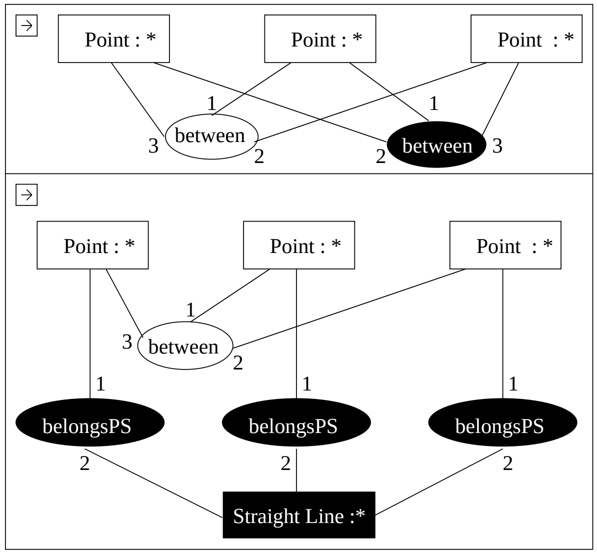

Figure 7: Representation of Axiom 2-1.

2.2.3 The composite concepts and relationships

From the basic objects and relationships defined in the membership's axioms and the order's axioms, HILBERT introduces composite concepts such as *Segment*, *Triangle* or *Polygon* and composite relationships such as *Same Side* or *Aligned*. These composite objects are represented by Concept/Relation Type Definitions. Figure 8 presents the definition of the concept *Triangle*. Figure 9 presents the definition of the relationship *Not Same Side*. Note that some of these definitions have to be completed with CG constraints in order to avoid erroneous deductions.

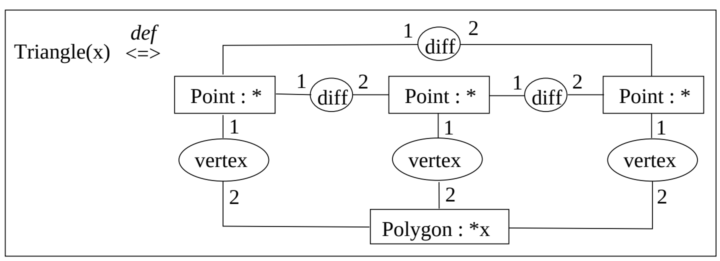

Figure 8: Representation of the composite concept called *Triangle*.

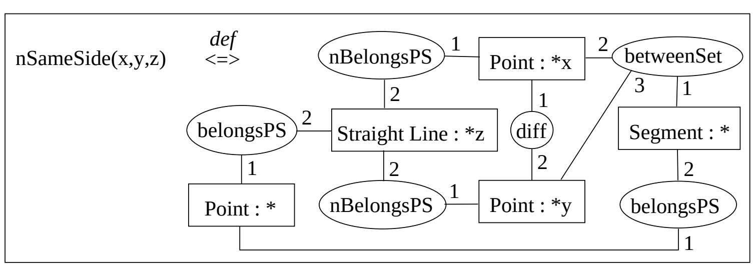

Figure 9: Representation of the composite relationship called *Not Same Side*.

2.3 The assertional level

The assertional level is used to represent facts. These facts are described with conceptual graphs which, themselves, are defined according to the conceptual vocabulary of the terminological level. A fact can be related to a theorem, a query of the end-user, a description of a geometric scene, etc. Figure 10 presents the representation of following query: is there exists a point which belongs to both an affine curve and a set of points ?

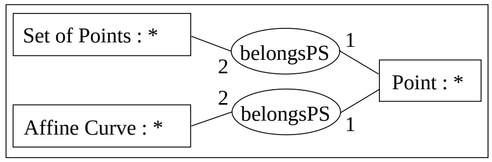

Figure 10: A graph representing a query.

The reasoning mechanisms provided by the CG model allow managing different types of exploitation of an ontology. In this section, we present two types of such an exploitation: the query answering and the theorem proving.

2.3.1 The query answering

The query answering consists in verifying if an assertion (represented by a conceptual graph) is true in a particular situation. Within the CG model, such an activity is performed by using the projection. Figure 11 presents an example of a projection from the query of figure 10 into the graph G representing a particular scene. In this context, the answer of the query is positive because there exists at least a projection: the point B belongs to both the straight line AB (which is an affine curve) and the straight line BC (which is a set of points).

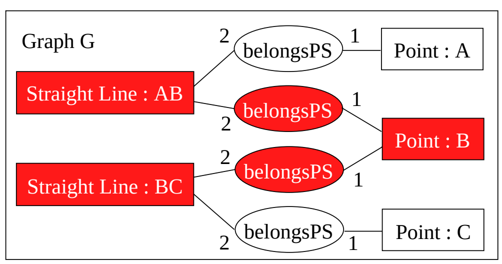

Figure 11: Query answering by using the projection.

2.3.2 The theorem proving

The theorem we consider in this section is expressed (in natural language) as follows: *There is at most one point which belongs to both a plan α and to a straight line d which does not belong to α.* This theorem is represented by the negative constraint of the figure 12.

The proof of this theorem expressed in natural language is as follows: « Suppose that two different points A and B belong to a plan α and to a straight line d which does not belong to α. As assumed by the axiom 1-6, all the points of d belong to α. So d belongs to α. This conducts to an absurdity ».

By using the CGs formalization, this proof is performed as follows.

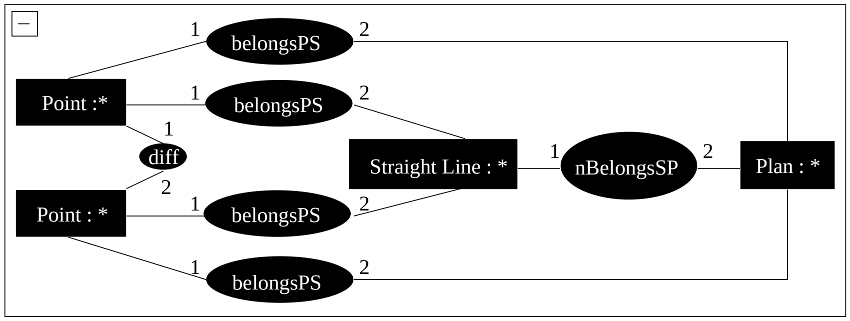

Figure 12: Representation of theorem 1.

The hypothesis of the proof (« Suppose that two different points A and B belong to a plan α and to a straight line d which does not belong to α ») is represented by the graph of the figure 13.

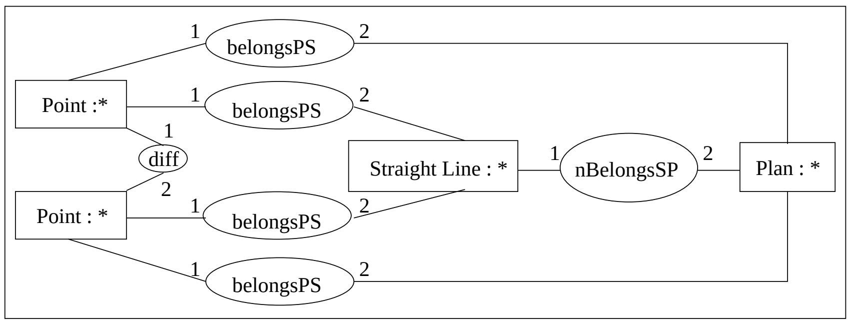

Figure 13: Representation of the hypothesis.

The first step of the proof (« As assumed by the axiom 1-6, all the points of d belong to α. So d belongs to $alpha$. ») is done by applying the rule used to represent the Axiom 1-6 (cf. figure 4).

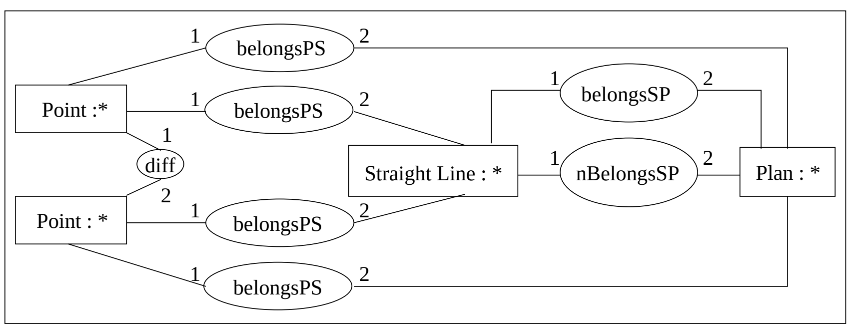

Figure 14: Application of Axiom 1-6.

The new graph (cf. figure 14) is not logically valid because he breaks the constraint which expressed the incompatibility of membership and no membership relationships (cf. figure 6). So, the graph which represents the hypothesis of the proof is a negative constraint.

3 Conclusion

From a methodological point of view, the ontology of the projective geometry we propose has been constructed by adopting a classical ontology engineering process. This process consists in first elaborating a corpus. Then, this corpus is analysed in order to identify the concepts and the relationships of the studied domain. This analysis can possibly be done by using tools related to natural language processing. In our work, all the concepts and their relations have been identified « by hands » from the particular corpus corresponding to the HILBERT's book. It is important to notice that most of the implicit knowledge has been identified when performing the activity of validation which, in our work, corresponds to the theorem proving.

From a practical point of view, the current version of the ontology is composed of 14 rules dedicated to the axioms, 32 rules dedicated to implicit knowledge, 3 negative constraints and 11 type definitions. This ontology is currently being implemented with the COGITANT framework [GENEST 1997].

References

[BACHIMONT 2001] B. BACHIMONT. *Modélisation linguistique et modélisation logique des ontologies : l'apport de l'ontologie formelle*, Actes des journées francophones d'Ingénierie des Connaissances, pp 349-368, Presse Universitaire Grenobloise (PUG), 2001.

[FENSEL & al. 2000] D. FENSEL & AL. *OIL in a nutshell*, Proceedings of European Knowledge Acquisition Workshop (EKAW'2000), LNAI n^o1937:1-16, Springer-Verlag, 2000.

[GENEST 1997] D. GENEST AND E. SALVAT. *A Platform allowing typed nested graphs: how CoGITo became CoGITaNT*, Proceedings of International Conference on Conceptual Structures (ICCS'1998), LNAI n^o1453:154-161, Springer Verlag, 1998.

[GRUBER 1993] T. R. GRUBER. *Toward principles for the Design Of Ontologies used for Knowledge Sharing*, in Formal Ontology in Conceptual Analysis and Knowledge Representation (GUARINO, POLI). Kluwer Academic Publishers, 1993.

[GUARINO & al. 2000] N. GUARINO AND C. WELTY. *A formal ontology of properties*, Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management, LNCS, Springer Verlag, 2000.

[HENDLER 2001] J. HENDLER AND D.L. MCGUINNESS. *The Darpa Agent Markup Language*, IEEE Intelligent Systems, 2001. http://www.daml.org.

[HILBERT 1997] D. HILBERT. *Les fondements de la Géometrie*, traduction of the book intitled *Grundlagen der Geometrie*. Editions Jacques Gabay, 1997.

[LECLERE 1997] M. LECLERE. *Reasoning with type definitions*, Proceedings of International Conference on Conceptual Structures (ICCS'1997), LNAI n^o1257:401-415, Springer Verlag, 1997.

[MUGNIER 2000] M.L. MUGNIER. *Knowledge representation and reasonings based on graph homomorphism*, Proceedings of International Conference on Conceptual Structures (ICCS'2000), LNAI n^o1867, Springer Verlag, 2000.

[RDFS 2000] RDFS, Resource Description Framework Schema Specification 1.0, http://www.w3.org/TR/2000/CR-rdf-schema-20000327/, 2000.

[SALVAT & al. 1996] E. SALVAT AND M.L. MUGNIER. *Sound and complete forward and backward chaining of graph rules*, Proceedings of International Conference on Conceptual Structures (ICCS'1996), LNAI n^o1115:248-262, Springer-Verlag, 1996.

[SOWA 1984] J. SOWA. *Conceptual Structures - Information Processing in Mind and Machine*, Addison-Wesley, Reading Mass, 1984.