A Generic Ontology for Spatial Reasoning

Frans Coenen and Pepijn Visser
CORAL*Research Group
Department of Computer Science, The University of Liverpool
Liverpool, UK

Abstract

In this paper we describe a generic ontology to support N-dimensional spatial reasoning applications. The ontology is intended to support both quantitative and qualitative approaches and is expressed using set notation. Using the ontology; spatial domains of discourse, spatial objects and their attributes, and the relationships that can link spatial objects can be expressed in terms of sets, and sets of sets. The ontology has been developed through a series of application studies. For each study a directed application ontology was first developed which was then merged into the generic ontology. Application areas that have been investigated include: Geographic Information Systems (GIS), noise pollution monitoring, environmental impact assessment, shape fitting, timetabling and scheduling, and AI problems such as the N-queens problem.

1 Introduction

In this paper we describe an generic ontology for spatial reasoning applications. The aim is to produce a mechanism which will facilitate the high level conceptualization of such applications in a manner which is both simple and as widely applicable as possible using ontologies. As a consequence researchers and manufacturers of spatial reasoning systems will benefit from the advantages that ontologies offer as demonstrated in other fields, for example Knowledge Based System (KBS) development [9].

The ontology is expressed using well established set theoretic notation. This approach was adopted because spatio-temporal objects naturally lend themselves to definition in terms of sets of addresses (coordinates). The ontology thus provides a mechanism whereby a spatial domain of discourse, spatial objects and their attributes, and the relationships that can link spatial objects can be expressed in terms of sets, and sets of sets. Application directed task ontologies can then be expressed using a subset of the available generic conceptualization.

A spatial reasoning system, the SPARTA (SPAtial Reasoning using Tesseral Addressing) system, which operates using a constraint logic programming paradigm, has also been developed. This system is designed to read directed problem

*Conceptualization and Ontology Research at Liverpool statement expressed using the generic ontology described here, and then to "run" the conceptualization so as to produce appropriate results. Although this system is referred to within this paper its precise nature is not described (interested readers are directed to Coenen et al. [5, 6] for further information).

We commence by presenting an introduction to ontologies (section 2) and the application area of spatio-temporal reasoning (section 3). Specific application ontologies, using the general ontology, are expressed using set notation. The advantages offered by this representation and some details with respect to syntax are thus presented in section 4. In section 5 some assumptions regarding the nature of multi-dimensional space and its definition with respect to the ontology are established. This is followed in section 6 and section 7 with discussion concerning the expression of spatial entities and the attributes that may be associated with such entities using the ontology. Finally in section 8 we present some conclusions and consider directions for further research into the field of spatial ontologies.

2 Ontologies

Building knowledge systems involves the creation of a model of a particular domain (e.g., electronic circuits, legislation, plants). Such a model is necessarily an abstraction of the domain under consideration. This is what makes a model useful, it abstracts from irrelevant details and thereby allow us to focus on the aspects of the domain we are interested in. Building a model of a domain involves deciding upon what entities in the domain are to be distinguished, and what relations will be recognized between these entities. Moreover, it involves deciding upon what types of entities, and what types of relation exist. Often, the latter kind of decisions are straightforward and not always explicitly documented. For instance, in building a model of the blocks world we use predicates like block(A), holds(A) and on(A,B) thereby implicitly assuming the domain to consist of blocks, tables, and hands, and that blocks, tables and hands have spatial relations. Making these assumptions requires the domain to be carved up in concepts. Alternatively stated, they reflect a conceptualization of the domain under consideration. The conceptualization tells us the types of entities and how they relate. In the blocks world the conceptualization tells us that there are blocks, tables, hands and that blocks and tables have spatial relations. It does not, however, tell us what blocks, tables and hands there are, nor how they are spatially related. To be able to compare conceptualizations of a domain, it is necessary to make them explicit. Thus, we need to have an explicit conceptualization. This is what the word ontology is used for in AI research: an ontology is defined as an explicit specification of a conceptualization ([8]).

Recently the result of gathering vocabularies and structuring domains has been recognized as a valuable effort in its own right, deserving of attention (Wiederhold p.7 [18]). In general, we can say that ontologies may contribute to the following five areas:

1. **Knowledge acquisition**. Ontologies describe and structure the entities

and relations that need to be acquired for the domain under consideration. Examples of this kind of ontology use are CUE [10], and MOBAL [14].

2. **Knowledge exchange**. Ontologies can be used to define assumptions that enable knowledge exchange between different systems. Examples of this kind of ontology use can be found in SHADE [11], and also in the work of Wiederhold [18]
3. **Knowledge-system design**. Ontologies are reusable constructs in the design of knowledge systems because they can be used to represent the invariant assumptions underlying different knowledge bases in the same domain. As such, they can be considered as initial building blocks of the knowledge base under construction. An example of this kind of ontology use can be found in the GAMES methodology [9].
4. **Domain-theory development**. Because an ontology explicitly states the building blocks of particular domains, it can be used for the analysis, comparison, and development of domain theories. Examples of this kind of ontology use can be found in Sim and Rennels [15], and Visser and Bench-Capon [17].
5. **System documentation**. Ontologies provide a meta-level view (vocabulary, structure) on their application domain which facilitates adequate system documentation for end-users. An example of this kind of ontology use is found in the Cyc project [13].

Although the ontologies concept is now well established in the Knowledge Based Systems (KBS) community ([9]) very little work has been undertaken in the field of spatial ontologies. Some work has been done on the language of spatial reasoning, i.e. "what is meant by the term nearby?" Etc. In addition mereology has been taken up as a theory of part-whole relations in formal philosophical ontology ([16]), although this has not been generally extended to spatial reasoning contexts. The ontology described here seeks to lay the foundation for a generic abstract ontology to support spatio-temporal reasoning applications so that the researchers and builders of such systems can gain similar advantages to does gained by the KBS community through the use of ontologies.

3 SPATIAL-TEMPORAL REASONING

Spatial reasoning can be defined, very broadly, as the automated manipulation of data objects belonging to spatial domains so as to arrive at application dependent conclusions. A spatial domain, in this context, is then considered to imply any N-dimensional space (real or imaginary, including 1-D temporal spaces) in which objects of interest may be contained. Automated manipulation then implies the computer simulation of some higher mental processes (not just the simple response to some stimulus or the mechanical performance of an algorithm). We are therefore not concerned with (say) the automated retrieval of spatially referenced data contained in some database format. Thus, at its simplest, spatial reasoning can be considered to revolve round the identification of (previously unknown) relationships that exists between spatial objects according to the relationships that are known or desired to exists between such data objects. In more complex systems the identified relations are then used as the foundation whereby further reasoning can take place, and consequently additional conclusions drawn.

4 Syntax

A number of languages/tools are available in which ontological conceptualizations may be expressed. The most well known example is Ontolingua [8] which is directed at ontologies for the conceptualization of expert systems and developed as part of the KSE project. Another example is KRSL (Knowledge Representation Specification Language) which is directed at a similar domain (Leher [12]).

The core ontology described here is presented using set notation. The principal reason for this is that space is conceptualized as a comprising groups of "cells" — the Cartesian coordinate system is the most widely used mechanism for referencing such cells. It is thus a natural step to consider such groups of cells in terms of sets. Further reasons for using set notation are that the theory is long established and is consequently well understood, and secondly that there is a recognized notation for expressing sets and the operations that can be performed on it.

Using the proposed spatio-temporal ontology a task dependent application ontology is described in terms of a "singleton" set, $TaskOntology$, which comprises a 3-tuple each element of which, in turn, is a set:

$$TaskOntology = \{\langle Space, Objects, Relations\rangle\}$$

The set *Space* describes the nature of the domain of discourse. The set *Objects* defines the set of spatial entities of interest, and the attributes that may be associated with those entities. The set *Relations* then defines the relationships that can exist between pairs of spatial entities. The more precise nature of each of these sets is described in the following sections.

5 Space

The ontology assumes that:

- N-dimensional space comprises a set of N-dimensional isohedral cells each of which is identified by a unique *address* (the SPARTA system uses a tesseral addressing system — see Diaz and Bell [7] for an overview of this technique).

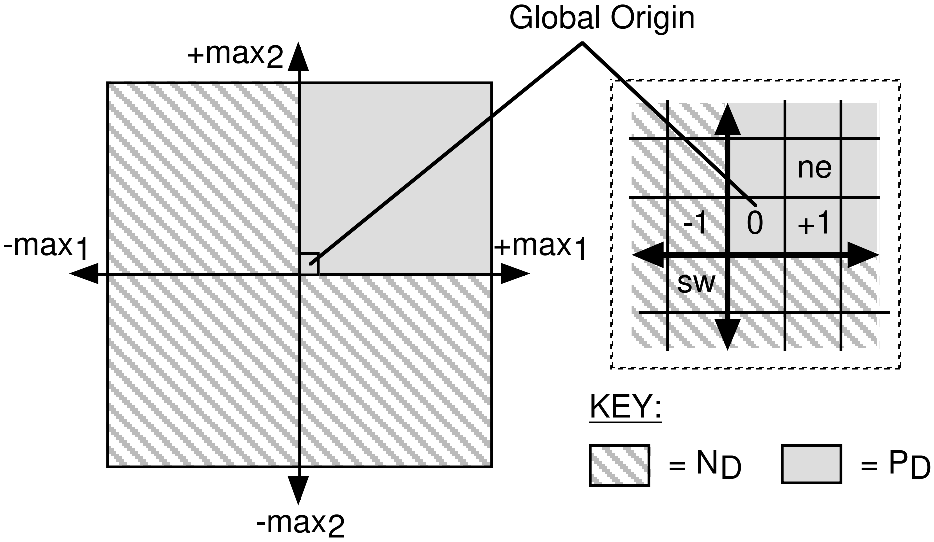

Figure 1: Example 2D space

- The addressing systems has the effect of linearising space so that an ordering can be imposed on sets of addresses (regardless of the number of dimensions under consideration).
- The addressing system has an origin to which all other addresses can be referenced
- The system has an arithmetic superimposed that facilitates translation through the space

Given this representation we can identify a domain of discourse U which may be defined as follows:

$$U = P \amalg N$$

(the disjoint union of the sets P and N) where:

$$P = \{p \mid p : (\textit{the set of all referencesrepresenting positive space})\}$$
$$N = \{n \mid n : (\textit{the set of all referencesrepresenting negative space})\}$$

The reasons for this distinction will become apparent when we go on to consider the attributes that can be associated with spatial objects in section 6.

An example 2-D space is given in Figure 1. Note that the maximum dimensions are denoted by the labels max_1 and max_2. The nature of these dimensions will be domain dependent. Note also that the *globalOrigin* of the space, the cell 0, is located at the intersection of the two axis. For the purposes of this document it is necessary to identify a number of further addresses with respect to the *globalOrigin*. These are shown in the inset in Figure 1 which assumes a linearisation the flows from the bottom-left cell up to the top-right cell. The addresses -1 and $+1$ indicate the addresses immediately before and after the *globalOrigin* as defined by the linearisation. The addresses sw and ne indicate the addresses which are "corner adjacent" to the *globalOrigin* but furthest away (in the negative and positive directions) with respect to the linearisation. The values that may be associated with the variables sw and ne will be implementation dependent.

The domain of discourse of any task is then either equivalent to U or is some subset of U (but such that the origin is included). In this paper we will indicate task domain sets using the notation U_D which is then comprised of the disjoint intersection of the positive and negative component spaces (P_D and N_D respectively).

Given a particular application the set *Space* will be defined as follows:

$$Space = \{max_1, max_2, ...max_N\}$$

where N is an integer representing the number of dimensions under consideration. Thus, in the context of a particular task ontology, a 1-D space would be defined as $\{max_1\}$, a 2-D space as $\{max_1, max_2\}$, and so on. This in turn will define the nature of the sets U_D, P_D and N_D. Once the domain of discourse has been defined we can define sub-spaces within this domain in terms of subsets of U_D, as dictated by the nature of the application.

5.1 Local Origins

We have seen that any set U_D has a *globalOrigin*. Similarly any subset of U_D will have a *localOrigin* which is defined as the reference nearest to the *globalOrigin* as dictated by the nature of the linearisation. Thus, given a subset X of the set U_D we define a function $f_{localOrigin}$ that returns the reference for the local origin of the set X. This function is defined as follows:

$$Dom(f_{localOrigin}) = Pow(U_D)$$
$$Cod(f_{localOrigin}) = U_D$$
$$Gr(f_{localOrigin}) = \{\langle X, a\rangle \mid a : X \wedge a\ nearest\ the\ (global)\ origin$$

where *Dom*, *Cod* and *Gr* are the domain, codomain and graph of the indicated function, and *Pow* is a function that returns the power set its argument (which must be a set).

5.2 Linearisation and sequences of addresses

The linear representation assumed offers many computational advantages. In particular it allows for the unambiguous identification of local origins and the definition of sequences of addresses. With respect to the latter we use the infix operator .. to define such sequences. The prefix operand for this operator then represents the start of the sequence (the address closest to the origin) and the postfix operand the end of the sequence (the address furthest away from the origin).

6 Objects

Any spatial application will comprise a number of spatial entities. These may describe points, intervals, events, areas, volumes and so on. Using the general ontology spatial entities of interest are described in terms of a set of sets, $Objects$, which is defined as follows:

$$Objects = \{Obj_1, Obj_2, ..., Obj_N\}$$

where the elements Obj_1 through to Obj_N are themselves set of sets whose elements describe attributes that may be associated with spatial entities. Spatial entities may have many attributes. The following are currently addressed by the general ontology:

- Identifier
- Location
- Shape
- Rotation
- Size (actual, maximum, minimum)
- Connectivity (a spatial entity does not have to be comprised of a continuous set of cells)
- Class (type)

This list of possible attributes is not complete, although it does cover those attributes most commonly encountered. Work is currently in progress to identify a more complete list. Each of the above attributes is discussed further in the following subsections.

6.1 Identifier

To do anything with a spatial entity we need to be able to identify it in some manner. The simplest approach is to give the entity in question a unique *name* selected from a set $Names$:

$$Names = n \mid the\ set\ of\ all\ possible\ names$$

Thus one (and only one) element of any set Obj is always a member of the set $Names$.

6.2 Location

Other than an identifier the second most important attribute that may be associated with a spatial entity is its location. There are two possibilities, either an entity's location is precisely known (i.e. it is fixed) or we know of some *location space* in which the entity can be said to exist (this may extend to the entire domain). What ever the case the location or location space is described in terms of some subset (*Location* or *LocationSpace*) of P_D, i.e. in terms of a positive set of cells contained within the domain. In the absence of any other information these sets are assumed to be equivalent to P_D. Thus:

$$\begin{aligned} \mathit{Location} &= \{L \mid L \subseteq P_D\} \\ \mathit{LocationSpace} &= \{L \mid L \subseteq P_D\} \end{aligned}$$

the definitions are the same, the interpretations are different. Note also that, given a spatial entity whose location is known (i.e. there exists a set *Location* as opposed to a set *LocationSpace*), further attributes such as shape, size, connectivity etc., are defined by default. We will refer to such an object as a *fixed object*.

6.3 Shape

Shape is probably the third most significant attribute that a spatial entity has. We have already seen that in the case of a fixed object shape is defined by default. However, in the absence of a fixed location, shape is often significant. Shape is defined (in a similar manner to the *Location* and *LocationSpace* sets) as follow:

$$\mathit{Shape} = \{S \mid S \subseteq U_D \wedge \mathit{globalOrigin} \in S\}$$

Note that any shape definition must include the origin address of U_D. This is required because if we wish to do anything useful with the shape it must have some reference (a local origin) with respect to which it can be manipulated.

We will refer to spatial entities that have a known shape, but no definite location other than some location space within which they are considered to exist, as *free objects* (in the sense that they are not fixed). Given a free object, the shape definition also defines further attributes such as size and connectivity. Note also that a free object must have some location space associated with it — somewhere within which it is known to exist. This may be either the entire space P_D or some subset of P_D as defined in the previous sub-section.

6.4 Rotation

A further important attribute associated with free objects is orientation — this may be either fixed or we may be free to rotate the object. Whether a shape/object can be rotated or not is defined in terms of a singleton set *Rotation*:

$$Rotation = \{r \mid r \in Bool\}$$

where *Bool* is the Boolean set comprised of the elements $\{true, false\}$.

6.5 Size

In the absence of location and shape information we may have knowledge of size. The size of a spatial object is expressed in terms of the number of members of a set describing its shape, i.e. the *cardinality* of this set. In the case of a fixed object this will be the set *Location*, in the case of a free object this will be the set *Shape*. Where the nature of this set is not known we can still express the required number of elements for this set. The minimum size of any object (that can be physically realized) is 1 indicating that it is represented by a single cell. The maximum size of an object is equivalent to $f_{cardinality}(P_D)$ (where $f_{cardinality}$ is a function that returns the "size" of its argument which must be a set), other wise the object exceeds the application domain. In addition the nature of the size of an object can be expressed in terms of (i) a minimum size, (ii) a maximum size, or (iii) an actual size — the set of operators $\{<, =, >\}$. Thus we can define a set *Size*, comprising a single 2-tuple, as follows:

$$Size \quad = \quad \{\langle q, z\rangle \mid q \in \{<, =, >\} \cdot z \in \{1 \mathinner{..} cardinality(P_D)\}\}$$

6.6 Connectivity

Finally we may know something about whether an object's location/shape is represented by a contiguous set of addresses or not. In this context continuity is defined as the situation where each element of a *Location* or *shape* set of addresses associated with any object is adjacent to at least one other element of this set (by adjacent we mean either edge or corner adjacency). We define the nature of the connectivity attribute in terms of a singleton set *Contiguity* as follows:

$$Contiguity = \{c \mid c \in Bool\}$$

6.7 Class

From the above we can conclude that (so far) the attributes of any object are taken from the set of all possible attributes (*Attributes*) defined as follows:

$$\begin{aligned} Attributes = \{&identifiers, locations, locationspaces,\\ &shapes, rotation, size, connectivity\}\end{aligned}$$

Given any spatial object only a sub-set of this possible set of attributes will be associated with it. The nature of this subset is, in turn, described by the *classes* of an object. This is defined by a singleton set $Class$:

$$Class = \{c \mid c \in \{fixed, free, shapeless\}\}$$

Thus the complete *Attribute* set should more properly be expressed as follows:

$$\begin{aligned} Attributes = \{&identifiers, classes, locations, locationspaces, \\ &shapes, rotation, size, connectivity\}\end{aligned}$$

The nature of the attributes that can be associated with each class has been discussed in the foregoing, however, a summary is presented below.

- **Fixed objects**: Objects whose locations are "fixed". Consequently they have only three attributes: an identifier, a class and a *Location* set (all other attributes are defined by default). Thus given a particular fixed object this will be defined as follows:

  $$FixedObject = \{name, class, Location\}$$

- **Free objects**: Objects whose location is not known, but who have a fixed shape. Objects belonging to this class have five attributes: an identifier, a class, *LocationSpace* and *Shape* sets, and whether rotation is permitted or not (the size and connectivity attributes are implied by the shape attribute and therefore do not need to be explicitly included). Thus given a specific free object this would be defined thus:

  $$FreeObject = \{name, class, LocationSpace, Shape, rotation\}$$

- **Shapeless objects**: Objects whose location and shape are both not known. These may be objects for which we know nothing (other than they exist), or we may know details about size and/or connectivity. Their definition will also require a description of the location space in which they are known to be contained. Thus:

  $$\begin{aligned} ShapelessObject = &\{name, class, LocationSpace\} \vee \\ &\{name, class, LocationSpace, Size\} \vee \\ &\{name, class, LocationSpace, Contiguity\} \vee \\ &\{Name, Class, LocationSpace, Size, Contiguity\}\end{aligned}$$

7 Relations

In addition to the above attributes, applicable to individual objects, there are a further set of attributes that serve to link pairs of objects or objects and locations. We will refer to such attributes as relationships. For example, given two objects we might say that one is contained within the other. The relation "contained" as used here is a predicate in the sense that the relation is either true or false. More precisely we are saying that the location (or locationSpace) of one of the objects is such that it is contained wholly within the location (or location space) of the other. We have seen that in the generic ontology, as outlined so far, all locations and location spaces are expressed as sets. Therefore the most natural way of describing relations in the context of the ontology described here is in terms of the standard set relations expressed as a Boolean functions:

$$
\begin{aligned}
&f_{filterEquals}(A,B) \Rightarrow true\ if\ A = B \\
&f_{filterIntersects}(A,B) \Rightarrow true\ if\ A \cap B \\
&f_{filterSubset}(A,B) \Rightarrow true\ if\ A \subset B \\
&f_{filterSubsetOrEquals}(A,B) \Rightarrow true\ if\ A \subseteq B \\
&f_{filterSuperset}(A,B) \Rightarrow true\ if\ A \supset B \\
&f_{filterSupersetOrEquals}(A,B) \Rightarrow true\ if\ A \supseteq B
\end{aligned}
$$

where A and B are sets of addresses. We use the term filter here to indicate that these relations are, in practice, used to express constraints that are desired to hold between objects. If we add the negation of these predicates we have 12 possible relations.

Alternatively we can also conceive of relations as non-Boolean functions which can be used to refine the location space associated with shapeless objects, i.e. given such an object we can get some idea of its nature through a set of relations. We can identify two such relations:

$$
\begin{aligned}
&f_{mapIntersects}(A,B) \Rightarrow A_n = A \cap B \\
&f_{mapComplement}(A,B) \Rightarrow A_n = B \setminus A
\end{aligned}
$$

We refer to these relations as mappings in the sense that we "map" some set operation (intersection, union, complement) on to the first argument (A) with respect to the second argument (B) so as the produce a revised set (A_n). Given a set of such relations we can continuously revise the location space for a shapeless object with respect to fixed or free objects until we have a desired location for the shapeless object.

The total possible set of relations is then as follows:

$$Relations \quad = \quad \{equals, intersects, subset, subsetEqual, superset,$$

$$supersetEqual, notEquals, notIntersects, notSubset,$$
$$notSubsetEqual, notSuperset, notSupersetEqual,$$
$$intersection, complement\}$$

In the ontology a relation, at its simplest, is expressed as a set of three elements

$$Relation \quad = \{\{a, r, b\} \mid a, b \in Names \cdot r \in Relations\}$$

Note that locations/location spaces are referenced using object names.

7.1 Offsets

To increase the expressiveness of these relations we can apply an *offset* to the location of an object. An offset is a set of addresses, defined as some subset of U_D, which is applied either to (a) all the elements describing a location (or location space) or (b) the local origin for this set. The effect in the first case is to translate or expand (or both) the space. In the second case this allows us to define other "locations" with respect to the given location or location space.

Although the offset principle allows us to expand objects the only way in which objects can be "shrunk" is to define a space with respect to the local origin of the object in question so that it appears that the object has been shrunk. This is often a cumbersome process, thus a facility is also provided to "shrink" by a given number of addresses.

What ever the case offsets are defined by a 2-tuple as follows:

$$Offset \quad = \quad \{\langle t, F\rangle | t \in \{ref, all, shrink\} \cdot F \subseteq U_D\}$$

Thus a relation can, more broadly, be defined as follows:

$$Relation = \{a, r, b\} \vee \{a, \langle t_a, F_a\rangle, r, b\} \vee$$
$$\{a, r, b, \langle t_b, F_b\rangle\} \vee \{a, \langle t_a, F_a\rangle, r, b, \langle t_b, F_b\rangle\}$$

7.2 Quantifying the intersection relations

The expressiveness of the *intersects* and *intersection* relations (either in addition to or as an an alternative to the use of offsets) may be further expanded by quantifying the size of the intersection. In the ontology we achieve this by allowing two optional arguments for the intersection predicate — an operator and a desired cardinality. The set of possible operators is as follows:

$$<, <=, =, =>, >$$

The cardinality is expressed as a positive integer in the range of 1 to $f_size(P_D)$. Thus we can (say) insist that an intersection is equal to a cardinality of 4 as follows:

$$\{a, intersection(=, 4), b\}$$

8 Conclusions

In this paper we have describe the foundation for an "all purpose" general purpose ontology to support the development of spatio-temporal reasoning applications. Although the technique has been applied to a great many applications — Geographic Information Systems (GIS) [4], noise pollution monitoring [2], environmental impact assessment [1], shape fitting [6], scheduling and timetabling [3], and AI problems such as the N-queens problem — the ontology is not yet complete. There are still many application areas, such as map and chart interaction, spatial simulation, environmental planning; which may require further mechanisms to increase the expressiveness of the relations. With respect to the attributes that can be associated with spatial entities there are still a number of variations on the given list that require further investigation. For example we can conceive of partially shaped objects in that we may know a minimum shape for the object. There is thus still much work to do. However, work to date, indicates that the ontology described in this paper provides a sound foundation for this further work.

References

[1] Beattie, B., Coenen, F.P., Hough, A., Bench-Capon, T.J.M., Diaz, B.M. and Shave, M.J.R. (1996). Spatial Reasoning for Environmental Impact Assessment. Third International Conference on GIS and Environmental Modeling, Santa Barbara: National Center for Geographic Information and Analysis, WWW and CD.

[2] Brown, A.G.P., Coenen, F.P., Shave, M.J. and Knight, M.W. (1998). An AI Approach to Noise Prediction. To appear in Building Acoustics.

[3] Coenen, F.P., Beattie, B., Bench-Capon, T.J.M., Shave, M.J.R and Diaz, B.M. (1995). Spatial Reasoning for Timetabling: The TIMETABLER system. Proceedings of the 1st International Conference on the Practice and Theory of Automated Timetabling (ICPTAT'95), Napier University, Edinburgh, pp57-68.

[4] Coenen, F.P., Beattie, B., Bench-Capon, T.J.M, Diaz, B.M. and Shave. M.J.R. (1996). Spatial Reasoning for Geographic Information Systems. Proceedings 1st International Conference on GeoComputation, School of Geography, University of Leeds, Vol 1, pp121-131.

[5] Coenen, F.P., Beattie, B., Bench-Capon, T.J.M, Diaz, B.M. and Shave. M.J.R. (1997). A tesseral Approach to N-Dimensional Spatial Reasoning.

In Hameurlain, A. and Tjoa, A.M. (Eds), Database and Expert Systems Applications, (Proceedings DEXA'97), Lecture Notes in Computer Science 1308, Springer Verlag, pp633-642.

[6] Coenen, F.P., Beattie, B., Bench-Capon, T.J.M., Diaz, B.M. and Shave. M.J.R. (1998). Spatio-Temporal Reasoning Using A Multi-Dimensional Tesseral Representation. Proceedings ECAI'98, pp140-144.

[7] Diaz, B.M. and Bell, S.B.M. (1987). Spatial Data Processing using Tesseral Methods. Natural Environment Research Council publication, Swindon, England.

[8] Gruber, T. (1992). Ontolingua: A Mechanism to Support Portable Ontologies. Knowledge Systems Laboratory, Standford University, Stanford, USA.

[9] Heijst, G. van (1995). The Role of Ontologies in Knowledge Engineering, Doctoral Thesis, University of Amsterdam, Amsterdam, The Netherlands.

[10] Heijst, G. Van, and G. Schreiber (1994). CUE: Ontology-based knowledge acquisition, L. Steels, A. Th. Schreiber, and W. Van de Velde (eds.), A Future for Knowledge Acquisition, Proceedings of the 8th European Knowledge Acquisition Workshop EKAW'94, Vol. 867 of Lecture Notes in Artificial Intelligence, pp.178-199, Springer-Verlag, Berlin/Heidelberg, Germany.

[11] Kuokka, D.R., J. McGuire, J.C. Weber, J.M. Tenenbaum, T.R. Gruber, and G.R. Olsen (1993). SHADE: Knowledge-Based Technology for the Re-Engineering Problem, Annual Report 1993.

[12] Leher, N. (1990). Knowledge Representations Specification Language. Technical Report, DARPA/Rome Laboratory Planning and Scheduling Initiative, Reference manual.

[13] Lenat, D.B. and R.V. Guha (1990). Building Large Knowledge-Based Systems; Representation and Reasoning in the Cyc Project, Addison-Wesley, Reading, Massachusetts, United States.

[14] Morik, K., S. Wrobel, J-U. Kietz, and W. Emde (1993). Knowledge Acquisition and Machine Learning; Theory, Methods and Applications, Knowledge-Based Systems, Academic Press Limited, London, United Kingdom.

[15] Sim, I., and G. Rennels (1995). Developing A Clinical Trial Ontology: Comments on Domain Modeling and Ontological Reuse, Knowledge Systems Laboratory Medical Computer Science, KSL-95-60, June 1995.

[16] Simons, P. (1987). Parts — A study in Ontology. Clarendon Press, Oxford.

[17] Visser, P.R.S. and T.J.M. Bench-Capon (1998). A Comparison of Four Ontologies for the Design of Legal Knowledge Systems, Artificial Intelligence and Law, Vol 6,N0 1, pp3-26.

[18] Wiederhold, G. (1994). Interoperation, Mediation, and Ontologies, Proceedings International Symposium on Fifth Generation Computer Systems (FGCS94), Workshop on Heterogeneous Cooperative Knowledge-Bases, Vol. W3, pp.33-48, ICOT, Tokyo, Japan.