Published in: L. Padgham et al., editors, Proceedings of the International Workshop on Description Logics, Nov. 2-4, 1996, Cambridge, Massachusetts, AAAI Press, Menlo Park, California, 1996, Technical Report No. WS-96-05, pp. 124–128.

Using Description Logic for Reasoning about Diagrammatical Notations

Volker Haarslev
University of Hamburg, Computer Science Department
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany
`http://kogs-www.informatik.uni-hamburg.de/~haarslev/`

1 Introduction

This paper summarizes research [1; 2; 3] about a fully implemented logical framework to develop axiomatizations defining meaningful "constellations" of *abstract* diagrammatical objects. The proposed framework is based on a spatial logic for describing qualitative spatial relationships between objects and on description logic (DL) as specification formalism. The framework was successfully applied to three representative diagrammatic notations: simple entity-relationship (ER) diagrams, place-transition petri nets, and a visual language for concurrent logic programming.

This logical framework forms the basis for the generic object-oriented editor GenEd that supports the formal design and analysis of visual notations. Prominent features of GenEd are (1) it is *generic*, i.e. domain-specific syntax and semantics of drawings are specified as TBox expressions; (2) *built-in parser* for actual drawings that creates ABox assertions in accordance to our spatial logic; (3) *reasoning* capabilities about diagrams and their specification by utilizing the classifier and realizer of DL systems.

The next section sketches our theoretical foundation for this approach. It is followed by an example session applying our framework to entity-relationship diagrams and giving a short overview of GenEd's user interface. Afterwards we report on our experience with two major DL systems. We conclude this paper with a short discussion of related work and future research.

2 Theoretical Foundation

Our approach is based on a fully formalized theory [2] for describing visual notations that consists of three components. Each component is defined by precise semantics. Objects and relations are defined by point-sets and topology. DL can be based on model-theoretic semantics using a compositional axiomatization with set theory. GenEd implements these three components in accordance to our theory.

2.1 Geometrical Objects

The implementation of geometrical objects and recognition of spatial relations uses well-known computer graphics techniques for reasons of efficiency. The semantics of these algorithms are still specified within our theory (see [4] for a complete treatment). GenEd offers a set of

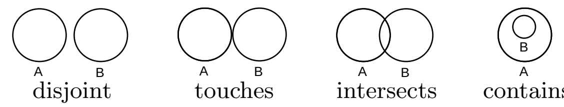


Figure 1: Primitive relations between A and B

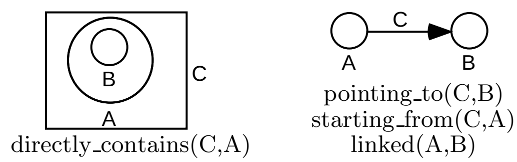


Figure 2: Higher-level relations

predefined geometrical objects (similar to other object-oriented graphic editors) that can be used to design examples of particular notations. Supported primitive objects are points, (directed) line segments, line segment chains, and (spline) polygons. These objects can be used to compose other objects (e.g. circles, ovals, etc).

2.2 Spatial Relations

GenEd recognizes seven primitive spatial relations (*disjoint, touches, intersects, contains/contained_by, covers/covered_by*) that may hold between objects (see Figure 1). It also computes the dimension of the intersection, if applicable. The semantics are defined in analogy to a proposal by Clementini et al. [5] and are based on point-sets and topology. The relations have a parameterized 'fuzziness' compensating for inexact positioning of objects (caused by users or scaling factors) and floating-point arithmetic. In contrast to several other approaches for spatial relations (e.g. see [2]) GenEd can also deal with concave objects. Additionally, an arbitrary collection of objects may be grouped together and treated as a *composition* object. Analogous semantics for composition objects were defined.

Higher-level relations were implemented with the help of the above mentioned seven relations (see Figure 2). GenEd currently recognizes specialized containment (directly-contains/inside), connectivity (linked-with), direction of line segments (starting-from, pointing-to), and partonomies (has-part/part-of). These relations are also applicable to composition objects.

2.3 Description logic

Description logic is used to combine geometrical objects and spatial relations. GenEd supports a predefined *upper model* resembling built-in geometrical objects and computed relations. Objects are represented as a hierarchy of (primitive) concept definitions and relations as a set of (primitive) role definitions also partly organized in a hierarchy. We consider visual notations as a subclass of formal languages. Language elements are specified as defined concepts that represent meaningful constellations of geometrical objects. We decided to bypass the aggregation problem and always select one geometrical object from a constellation that represents this aggregation with the has-parts role.

In the following we describe the semantics of our DL in the usual manner. Let $\mathcal{C}$ be the set of concepts and $\mathcal{R}$ the set of roles in a DL theory. A model is a set $\mathcal{D}$ and an assignment function ξ such that $\xi : \mathcal{C} \longrightarrow 2^{\mathcal{D}}$, $\xi : \mathcal{R} \longrightarrow 2^{\mathcal{D}^2}$ where $2^{\mathcal{D}}$ is the powerset of the domain $\mathcal{D}$, where $\mathcal{D}^2 = (\mathcal{D} \times \mathcal{D})$ and where ξ must satisfy the following conditions (concept names are denoted by c and role names by r):

$$
\begin{aligned}
\xi[\text{concept name}] &\subseteq \mathcal{D} \\
\xi[\text{role name}] &\subseteq \mathcal{D} \times \mathcal{D} \\
\xi[(c_1 \wedge \ldots \wedge c_n)] &= \cap_{i=1}^{n} \xi[c_i] \\
\xi[(c_1 \vee \ldots \vee c_n)] &= \cup_{i=1}^{n} \xi[c_i] \\
\xi[(\exists_{\geq \mathsf{n}}\ \mathsf{r})] &= \{x \mid \|\{(x,y) \mid (x,y) \in \xi[r]\}\| \geq n\} \\
\xi[(\exists_{\geq \mathsf{n}}\ \mathsf{r}\ \mathsf{c})] &= \{x \mid \|\{(x,y) \mid (x,y) \in \xi[r] \wedge y \in \xi[c]\}\| \geq n\} \\
\xi[(\exists_{\leq \mathsf{n}}\ \mathsf{r})] &= \{x \mid \|\{(x,y) \mid (x,y) \in \xi[r]\}\| \leq n\} \\
\xi[(\exists_{\leq \mathsf{n}}\ \mathsf{r}\ \mathsf{c})] &= \{x \mid \|\{(x,y) \mid (x,y) \in \xi[r] \wedge y \in \xi[c]\}\| \leq n\} \\
\xi[(\exists_{= \mathsf{n}}\ \mathsf{r})] &= \{x \mid \|\{(x,y) \mid (x,y) \in \xi[r]\}\| = n\} \\
\xi[(\exists_{= \mathsf{n}}\ \mathsf{r}\ \mathsf{c})] &= \{x \mid \|\{(x,y) \mid (x,y) \in \xi[r] \wedge y \in \xi[c]\}\| = n\} \\
\xi[(\forall\ \mathsf{r}\ \mathsf{c})] &= \{x \mid \forall y : (x,y) \in \xi[r] \Rightarrow y \in \xi[c]\} \\
\xi[(=\ \mathsf{r}\ \mathsf{i})] &= \{x \mid \forall y : (x,y) \in \xi[r] \Rightarrow y = \mathsf{i}\} \\
\xi[(\mathsf{r}\bullet\ \mathsf{c})] &= \xi[r] \cap \{(x,y) \mid y \in \xi[c]\} \\
\xi[\mathsf{r}_1 \circ \mathsf{r}_2] &= \{(x,y) \mid \exists z.(x,z) \in \xi[r_1] \wedge (z,y) \in \xi[r_2]\}
\end{aligned}
$$

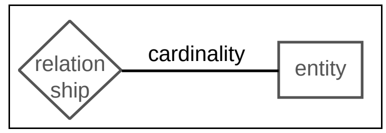
3 Example Session: ER Diagrams

We applied our framework to three exemplary diagrammatic notations: simple entity-relationship (ER) diagrams, state-transition petri nets, visual programming languages, and evaluated its feasibility in the domain of geographical information systems (see [1; 3] for more details).

3.1 Knowledge Base

In the following we outline the specification of ER diagrams. Our taxonomy of concepts is designed in a way that every element of an example drawing has to be classified as an instance of a concept that is a leaf in the taxonomy. Elements that violate this property are an indication for a bug occurring either in the example program or in the KB. Figure 3 shows a screen shot of GenEd's user interface displaying a subpart of a larger example modeling relationships in an airline company. We assume a few primitive concepts and spatial relations from GenEd's upper model that represent geometrical objects (rectangle, circle, diamond, line, text) used in our ER diagram language. In the following, primitive concepts are typeset in a slanted style.

Connectors

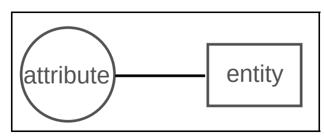


A *relationship-entity* connection is a line that touches exactly one text label (expressing cardinality) and two other regions (rectangle or diamond). A *cardinality* is a text string with values chosen from the set $\{1, m, n\}$.

relationship_entity $\equiv$
(*line* $\wedge$ ($\exists_{=3}$ touching) $\wedge$ ($\exists_{=1}$ touching *text*) $\wedge$
($\exists_{=2}$ touching (*rectangle* $\vee$ *diamond*)) $\wedge$
($\exists_{=1}$ touching *rectangle*) $\wedge$ ($\exists_{=1}$ touching *diamond*))

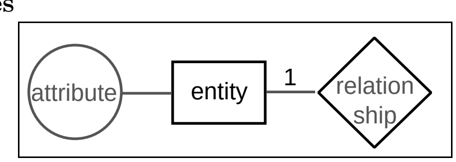
cardinality $\equiv$
(*text* $\wedge$ ($\forall$ touching relationship_entity) $\wedge$
($\exists_{=1}$ touching) $\wedge$ ($\forall$ text_value $\{1, \mathsf{m}, \mathsf{n}\}$))

An *attribute-entity* connection is a line that touches only two regions (circle or rectangle) and no text string.



attribute_entity $\equiv$
(*line* $\wedge$ ($\exists_{=2}$ touching) $\wedge$
($\forall$ touching (*circle* $\vee$ *rectangle*)) $\wedge$
($\exists_{=1}$ touching *rectangle*) $\wedge$ ($\exists_{=1}$ touching *circle*))

Entities



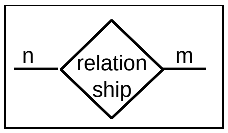
An *entity* is a rectangle that contains its name. It touches at least one relationship-entity and optionally some attribute-entity connectors. It is linked with at least one diamond.

named_region $\equiv$
(*region* $\wedge$ ($\exists_{=1}$ containing) $\wedge$ ($\forall$ containing *text*))

entity $\equiv$
$(\mathit{rectangle} \wedge \text{named\_region} \wedge$
$(\exists_{\geq 1}$ touching relationship_entity$) \wedge$
$(\forall$ touching (attribute_entity $\vee$ relationship_entity$)) \wedge$
$(\exists_{\geq 1}$ linked_with *diamond*$) \wedge$
$(\forall$ linked_with $(\mathit{circle} \vee \mathit{diamond})))$

Relationships

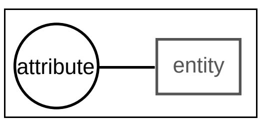
A *relationship* is a diamond that contains its name. It touches one relationship-entity and optionally some attribute-entity connectors. It is linked with two entities.



relationship $\equiv$
$(\mathit{diamond} \wedge \text{named\_region} \wedge$
$(\exists_{=2}$ linked_with$) \wedge (\forall$ linked_with entity$) \wedge$
$(\exists_{=2}$ touching$) \wedge (\forall$ touching relationship_entity$) \wedge$
$(\exists_{\leq 2}$ touching $(=$ (touching $\circ$ text_value) 1$)) \wedge$
$(\exists_{\leq 1}$ touching $(=$ (touching $\circ$ text_value) m$)) \wedge$
$(\exists_{\leq 1}$ touching $(=$ (touching $\circ$ text_value) n$)))$

Attributes

An *attribute* is a circle that contains its name. It touches one attribute-entity connector and is linked with an entity.



attribute $\equiv$
$(\mathit{circle} \wedge \text{named\_region} \wedge$
$(\exists_{=1}$ linked_with$) \wedge (\forall$ linked_with entity$))$

3.2 Implementation

GenEd is implemented in Common Lisp using the Common Lisp Object System (CLOS) and the Common Lisp Interface Manager (CLIM) as interface toolkit. The classification of concepts and individuals takes place by using the lisp implementation of Classic [6; 7] as DL system. GenEd consists of 28 modules with a total of about 300 KB source code (without CLIM, CLOS, and Classic).

3.3 User Interface

The general procedure for working with GenEd is as follows. The user loads a domain-dependent knowledge base (KB) into GenEd. This KB has to comply to GenEd's upper model. A new drawing may be created in the workspace (center window in Figure 3) or an existing one loaded. The built-in spatial parser analyzes a drawing in accordance to the upper model and creates ABox individuals and assertions resembling the elements of the drawing and their spatial relationships. Afterwards GenEd invokes the DL system. A protocol of the classification process can be displayed in GenEd's rightmost vertical window. GenEd optionally shows the concept membership of drawing elements and several other useful information (see center window).

GenEd supports two reasoning modes. While in *incremental* mode GenEd records differences to previous states and reports these differences to the ABox. The reasoning process is invoked to automatically analyze drawings after every modification and to give the user an immediate feedback. If the *batch* mode is set drawings are always analyzed from scratch and the user has to start the reasoning process manually.

4 Experience with DL Systems

The first prototype for our framework used Loom [8] (version 2.1) as DL system. The logic implemented by Loom is quite powerful and was sufficient to express static semantics of a visual language for concurrent logic programming (see [1; 9] for examples). In case of transitive or recursive roles we escaped the standard DL by using rules or Loom's 'satisfies' feature. It turned out that number restrictions for role fillers, role value maps, qualified roles, union of concepts (OR), and an implicit closed-word assumption (CWA) for selected concepts and relations are very important features of a DL suitable for the diagrammatic reasoning domain.

Our second and current prototype (GenEd) uses Classic (version 2.2) instead of Loom. A major advantage of Classic is its stable, bug-free implementation and the support of a fully implemented explanation facility for TBox and ABox reasoning. However, Classic implements only a subset of the logic supported by Loom. Classic allows only the definition of primitive roles and of role value maps that are restricted to attributes. It does not support qualified or domain/range restricted roles or the union of concepts. Closed-world reasoning is only available by explicitly closing roles for individuals. We also missed Loom's powerful query facilities about the state of its ABox.

Since we decided to 'live with Classic' we had to escape Classic's standard DL and to partially emulate or work around the missing features with the help of Classic's rule facility. The union of concepts was accomplished by the definition of a new primitive concept representing an 'OR-concept' and by a set of rules each associated with one 'OR-part' as trigger concept. A corresponding rule fires whenever an individual is classified as member of an OR-part and it asserts the 'OR-concept' membership for this individual. This solution was sufficient for our domain since we needed these OR-concepts mostly at the bottom of the subsumption tree. for instance, a term `(or circle diamond)` is replaced by `circle-or-diamond` as name for a new primitive OR-concept.

The second major problem of emulating qualified roles

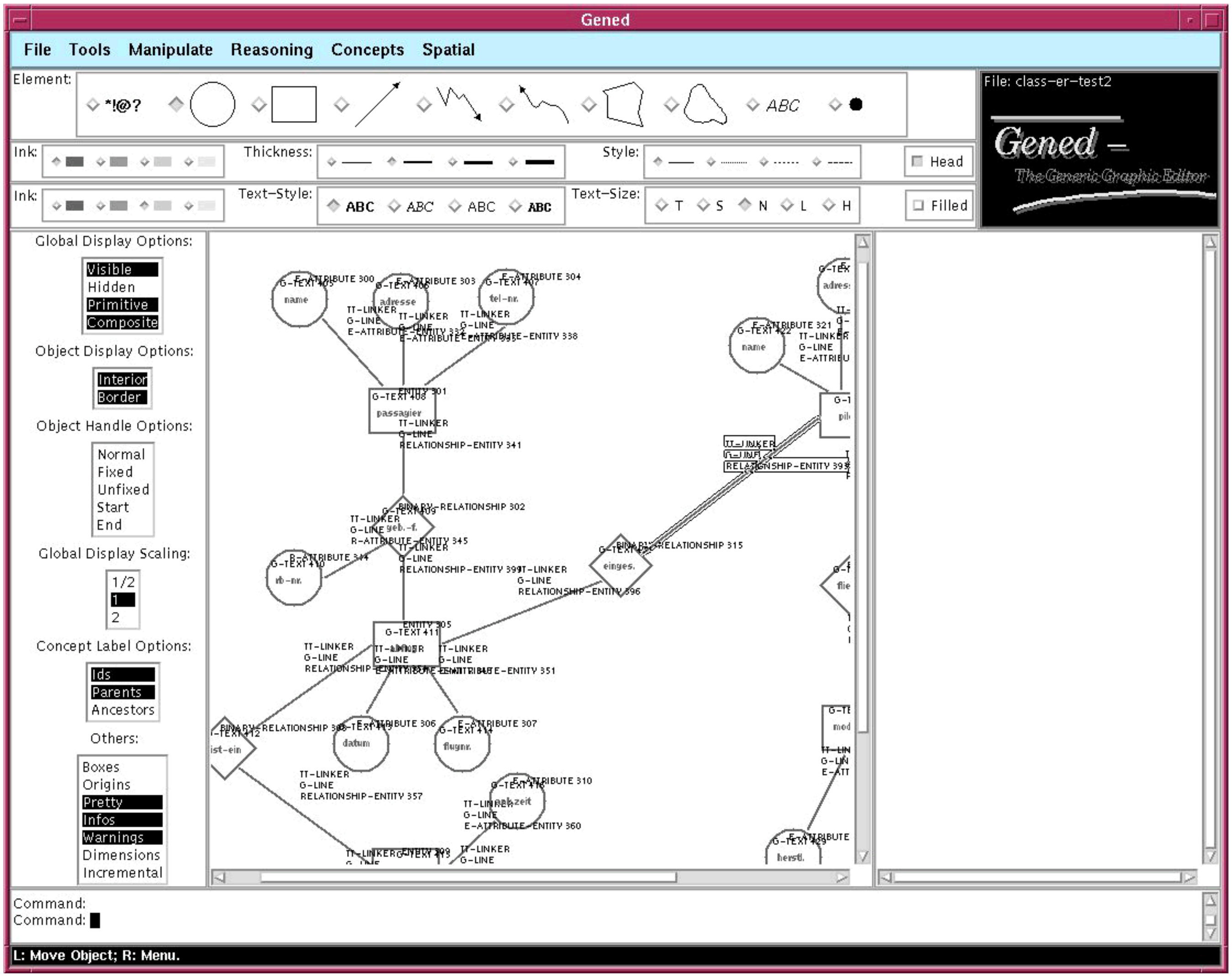


Figure 3: GenEd's user interface with an ER diagram modeling airlines

was much harder to solve. A range-restricted or qualified role is defined as a primitive subrole of a proper parent. The new subrole and its parent are required to have an inverse role since only the range restriction is known as trigger concept. Each new subrole is associated with a 'filler rule' that computes fillers for the inverse of this subrole. The filler rule is triggered whenever an individual is classified as member of the 'qualifying concept' and this individual is already known as filler of the parent's inverse of this new subrole. For instance, a term `(at-least 1 touching circle)` is replaced by `(at-least 1 touching-circle)` where `touching-circle` names the new primitive subrole.

The third problem is strongly related to the second one and caused by Classic's approach to closing of roles. Our domain depends heavily on inferences about 'all' and 'at-most terms' that are only possible with a CWA about the involved role and its fillers. Therefore, every role has to be closed for every known individual in order to get these inferences. Usually our solution for qualified roles results in a large set of subroles and worsens the time complexity of role closing. Moreover, the ordering of role closing is critical since filler rules may depend on the result of classifications triggered by other filler rules and role closing. In general, a top-down, breadth-first approach for closing the role hierarchy is safe if it complies to the implicit dependencies between subroles and their qualifying concepts.

5 Related Work

Our approach is mainly intended for the design and evaluation of diagrammatic visual notations/formalisms. We are not aware of any other approach using DL for diagrammatic reasoning. The understanding of diagrams can be considered as a subproblem of image interpretation and is related to similar approaches in this area. The first treatment in this area was the MAPSEE approach [10]. However, their specifications rely on first-order predicate logic and cannot gain from the advantages of our DL approach. We also argue that DL notation — featuring concept and role definitions with inheritance and with a possible extension to concrete domains— is much more suitable for human and even mechanical inspection. This is an important issue since visual formalisms are still designed by humans. Another approach for the logical reconstruction of image interpretation [11; 12] uses DL as framework. A survey of related work for visual languages can be found in [2].

6 Conclusion and Further Research

We presented a framework for diagrammatic reasoning about visual notations that is based on DL. We successfully applied this approach to specify formal semantics of several representative visual notations. We are currently integrating an approach [13] to combine object-oriented programming and DL by offering a generic CLOS layer for programming with Classic. We are planning to incorporate concrete domains over the algebra of simple reals, i.e. extending our framework for reasoning about systems of (in)equalities over (non)linear polynomials.

References

[1] V. Haarslev, "Formal Semantics of Visual Languages using Spatial Reasoning", in *1995 IEEE Symposium on Visual Languages, Darmstadt, Germany, Sep. 5-9*. Sept. 1995, pp. 156–163, IEEE Computer Society Press.

[2] V. Haarslev, "A Fully Formalized Theory for Describing Visual Notations", in *International Workshop on the Theory of Visual Languages, May 30, 1996, Gubbio, Italy*, May 1996.

[3] V. Haarslev and M. Wessel, "GenEd – An Editor with Generic Semantics for Formal Reasoning about Visual Notations", in *1996 IEEE Symposium on Visual Languages, Boulder, Colorado, USA, Sep. 3-6*. Sept. 1996, IEEE Computer Society Press, in press.

[4] M. Wessel, "Development of a concept-oriented Generic Graphic Editor in Common Lisp (in German)", Jan. 1996, Studienarbeit.

[5] E. Clementini, P. Di Felice, and P. van Oosterom, "A Small Set of Formal Topological Relationships Suitable for End-User Interaction", in *Advances in Spatial Databases, Third International Symposium, SSD'93, Singapore, June 23-25, 1993*, D. Abel and B.C. Ooi, Eds. June 1993, vol. 692 of *Lecture Notes in Computer Science*, pp. 277–295, Springer Verlag, Berlin.

[6] R.J. Brachman, D.L. McGuinness, P.F. Patel-Schneider, L.A. Reswnick, and A. Borgida, "Living with Classic: When and How to Use a KL-ONE-like Language", In Sowa [14], pp. 401–456.

[7] R.J. Brachman, ""Reducing" CLASSIC to Practice: Knowledge Representation Theory Meets Reality", in *Principles of Knowledge Representation and Reasoning, Third International Conference, Cambridge, Mass., Oct. 25-29, 1992*, Oct. 1992, pp. 247–258.

[8] R.M. MacGregor, "The Evolving Technology of Classification-based Knowledge Representation Systems", In Sowa [14], pp. 385–400.

[9] V Haarslev, R. Möller, and C. Schröder, "Combining Spatial and Terminological Reasoning", in *KI-94: Advances in Artificial Intelligence – Proc. 18th German Annual Conference on Artificial Intelligence, Saarbrücken, Sept. 18–23, 1994*, B. Nebel and L. Dreschler-Fischer, Eds. Sept. 1994, vol. 861 of *Lecture Notes in Artificial Intelligence*, pp. 142–153, Springer Verlag, Berlin.

[10] R. Reiter and A.K. Mackworth, "A Logical Framework for Depiction and Image Interpretation", *Artificial Intelligence*, vol. 41, pp. 125–155, 1989.

[11] H. Lange and C. Schröder, "Analysis and Interpretation of Changes in Aerial Images: Knowledge Interpretation and Spatial Reasoning", in *ISPRS Commision III Symposium – Spatial Information from Digital Photogrammetry and Computer Vision, Munich, Germany, Sep. 5–9, 1994*, H. Ebner, C. Heipke, and K. Eder, Eds., Sept. 1994, vol. 30 of *International Archives of Photogrammetry and Remote sensing*, pp. 475–482.

[12] C. Schröder and B. Neumann, "On the Logics of Image Interpretation: Model-Construction in a Formal Knowledge-Representation Framework", in *Proceedings of the 1996 IEEE International Conference on Image Processing ICIP-96, Lausanne, September 16-19, 1996*. Sept. 1996, IEEE Computer Society Press, in press.

[13] R. Möller, "A Functional Layer for Description Logics: Knowledge Representation Meets Objects-Oriented Programming", in *OOPSLA'96*, Oct. 1996, in press.

[14] J.F. Sowa, Ed., *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, San Mateo, California, 1991. Morgan Kaufmann Publishers.