# Spatial Reasoning with Description Logics

**Volker Haarslev**

University of Hamburg, Computer Science Department, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany, haarslev@informatik.uni-hamburg.de

## 1 Methodology

We outline a logical framework to develop axiomatizations defining meaningful "constellations" of *abstract* spatial objects. The proposed framework is based on a spatial logic for describing qualitative spatial relationships between objects and description logics theory (see also [1]). The framework was successfully applied to the domain of visual programming languages. These languages are represented by (at least 2D) graphics instead of text. Within our framework syntax and semantics of "completely" visual languages can be based on spatial relationships between their graphical language elements.

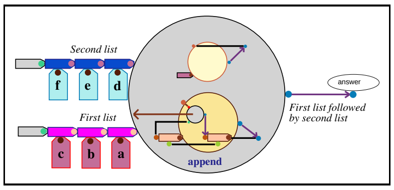


Figure 1: Append of two lists in PJ

In this context we express static semantics of visual languages as a set of axioms defining meaningful constellations of language elements. The specifications defining our example language, Pictorial Janus, are fully developed and were automatically verified using a representative set of example programs. The reader is referred to [2] for a complete treatment. Pictorial Janus [3] (PJ) is a completely visual language for the domain of concurrent logic programming. PJ's syntax is defined through topological relations which have to hold between language elements. PJ's operational semantics can be defined by graphical transformation rules. Figure 1 shows a simple PJ program.

We employ a modified version of a spatial logic originally developed by Clementini et al. [4]. With respect to PJ our spatial logic deals with geometrical objects such as points, (directed) lines, convex regions, and text. It supports seven binary relations (disjoint, touching, overlapping, crossing, equal, covering/covered-by, containing/inside) which may hold between these geometric objects. These relations are mutually exclusive and cover every possible case. The objects and their possible topological relationships are defined with point-set theory (see also [1]).

We specified PJ's language elements with predicates stating necessary and sufficient conditions. These definitions can be simplified by removing every cyclic dependency between them. These dependencies result from the original, informal definition of PJ [3] that is somewhat ambiguous and context-sensitive. The simplified specifications are then translated to description logic (DL) theory and processed by appropriate DL systems. The assertional reasoner of DL systems is used to verify the semantics specifications through parsing of example programs.

The verification of formal semantics specifications is an important issue since these specifications are usually designed by humans. We argue that sufficiently complex specifications most likely contain bugs or are even incomplete, if there exists no mechanical way to perform verification. The verification process works as follows. Specifications of formal semantics and corresponding representations of example programs are analyzed by the DL system. Our taxonomy of concepts is designed in a way that every element of an example program has to be classified as an instance of a special concept marking legal interpretations. Elements that violate this property are an indication for a bug occurring in the example program or in the specifications.

## 2 Implementation and Related Work

The specifications were developed and verified with the help of a fully implemented system (using Common Lisp) that integrates a DL component (Loom [5]) with a special object-oriented editor for PJ programs and an interpreter for textually encoded PJ programs. The editor offers implements functions that recognize spatial relationships in accordance with our spatial logic. Our test suite consists of a reasonable number of programs implementing simple (e.g. list operations such as append, reverse, etc.) as well as more complex tasks (e.g. queue management, simple bank accounts, etc). Successfully parsed example programs are automatically translated, textually encoded, and executed by our interpreter. The iterative process of developing formal semantics specifications and verifying them with example programs turned out to be very fruitful.

The most relevant work is the approach of Cohn and Gooday [6] which also proposes formal semantics for PJ. In contrast to the current state of our work, they already developed formal operational semantics in accordance to the graphical transformation rules of PJ. However, their specifications still rely on full predicate logic and cannot gain from the advantages of our DL approach. As far as we know they currently do not support mechanical verification processes. We also argue that the DL notation featuring concept and role definitions with inheritance is much more suitable for human and even mechanical inspection.

## References


[1] 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.

[2] V. Haarslev, "Formal Semantics of (Completely) Visual Languages", Technical Report, in preparation, 1995.

[3] K.M. Kahn and V.A. Saraswat, "Complete Visualizations of Concurrent Programs and their Executions", in *1990 IEEE Workshop on Visual Languages, Skokie/IL, Oct. 4-6, 1990*. Oct. 1990, pp. 7–14, IEEE Computer Society Press.

[4] 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.

[5] D. Brill, *Loom Reference Manual, Version 2.0*, Marina del Rey, 1994.

[6] A.G. Cohn and J.M. Gooday, "Defining the Syntax and the Semantics of a Visual Programming Language in a Spatial Logic", in *AAAI-94, Spatial and Temporal Reasoning Workshop*, 1994, Preprint.