

22. Juli 1991

DISS. ETH Nr. 9396

**«Visual Prolog»
An Object-Oriented Platform for
Interactive Modeling and Problem Solving**

**A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZÜRICH**

**for the degree of
DOCTOR OF TECHNICAL SCIENCES**

**presented by
CARLO MULLER
Dipl. Informatik-Ing. ETH**

**born July 18, 1960
citizen of Luxembourg**

**Accepted on the recommendation of
Prof. Dr. W. Schaufelberger, examiner
Prof. Dr. J. Nievergelt, co-examiner**

1991

19. 7. 91

W. Schaufelberger

Abstract

This dissertation presents «Visual Prolog», a software environment for interactive model-based problem solving, in which the problem-solving skills of a human expert are coupled with the data management and visualization capabilities of modern workstations. In this synergy of man and machine, the expert takes the creative part while the computer provides the external aids essential for successful human problem solving.

The users of «Visual Prolog» formulate a conceptual model of their problem domain, visualize the objects involved in a concrete problem instance, and interact with the objects by observing and manipulating graphical symbols on the screen. The domain model is expressed in an object-oriented formalism built on top of the language Prolog. The model specifies the correct behavior of an object in all contexts and guarantees that its visual representation is consistent with its internal state. Objects can be equipped with procedural and declarative knowledge that helps the users in their efforts to solve a problem.

The object system of «Visual Prolog» has been designed to meet the requirements of interactive object-oriented modeling, where, in contrast to object-oriented programming, the end users directly access the involved objects. The object system fully exploits the positive features of Prolog, including the conversational nature of the interactive Prolog environment, as well as the ability to represent knowledge in a symbolic human-readable form. The major drawback of conventional Prolog systems – the lack of structure in the database of rules and facts – is compensated by the support of class hierarchies based on single inheritance, and object hierarchies based on object nesting. Graphical mouse-driven browsers are provided for both hierarchies.

A ready-to-use class library implements the complete framework for the visualization of objects and the interaction with them. The classes defined in the model inherit their visual behavior from this library, so that the model can focus on the representation of domain knowledge and need not be concerned with the implementation details of the visual user interface.

The «Visual Prolog» implementation includes an interpreter and a compiler for Prolog, extended by modules, types and global variables. The extensions allow the direct access to the graphics toolbox and the operating system of the host computer. Most parts of «Visual Prolog» are written in this language.

Kurzfassung

«Visual Prolog» ist eine Software-Umgebung, die das interaktive und modellbasierte Lösen von Problemen unterstützt. Sie verbindet die intellektuellen Fähigkeiten eines menschlichen Experten mit den Möglichkeiten moderner Arbeitsplatzrechner, Daten zu verwalten und visuell darzustellen. Bei dieser Zusammenarbeit von Mensch und Maschine übernimmt der Experte die kreative Rolle. Der Computer stellt die externen Hilfsmittel bereit, die zum erfolgreichen Problemlösen durch den Menschen notwendig sind.

Die Benutzer von «Visual Prolog» formulieren zuerst ein konzeptuelles Modell ihres Anwendungsgebietes. Dann stellen sie die Objekte, die in einem konkreten Problem vorkommen, bildlich dar und arbeiten mit ihnen, indem sie grafische Symbole auf dem Bildschirm manipulieren und beobachten. Das Modell des Anwendungsgebietes ist in einem objektorientierten Formalismus, der auf der Sprache Prolog aufbaut, beschrieben. Es bestimmt das korrekte Verhalten eines Objekts in allen Situationen und garantiert, dass seine visuelle Darstellung mit seinem internen Zustand übereinstimmt. Objekte können mit prozeduralem und deklarativem Wissen, welches den Benutzern bei der Lösung des Problems hilft, ausgestattet werden.

Das Objektsystem von «Visual Prolog» wurde speziell für die interaktive, objektorientierte Modellierung entworfen. Im Gegensatz zur objektorientierten Programmierung haben die Endbenutzer hier einen direkten Zugriff auf die Objekte. Das Objektsystem nutzt die positiven Merkmale von Prolog voll aus. Dazu gehört die Eigenschaft der interaktiven Prolog Umgebung, Fragen der Benutzer im Dialog zu beantworten, aber auch die Fähigkeit, Wissen in einer symbolischen, vom Menschen lesbaren Form darzustellen. Der wichtigste

Nachteil konventioneller Prologsysteme, nämlich das Fehlen jeglicher Struktur in der Datenbank aus Fakten und Regeln, wird durch die Unterstützung von Klassen- und Objekthierarchien ausgeglichen. Die Klassenhierarchie basiert auf der einfachen Vererbung, die Objekthierarchie auf der Verschachtelung von Objekten. Beide Hierarchien können mit grafischen, mausbedienbaren Werkzeugen untersucht werden.

Eine schlüsselfertige Klassenbibliothek enthält die komplette Infrastruktur, die zur Visualisierung und zur interaktiven Bearbeitung der Objekte benötigt wird. Die Klassen, die im Modell definiert sind, erben ihr visuelles Verhalten von den Klassen dieser Bibliothek. Dadurch kann sich das Modell auf die Darstellung des Anwendungswissens beschränken, und muss sich nicht um die Einzelheiten der visuellen Benützerschnittstelle kümmern.

«Visual Prolog» enthält einen Interpreter und einen Compiler für die Sprache Prolog, welche durch Module, Typen und globale Variablen ergänzt wurde. Die Erweiterungen ermöglichen den direkten Zugriff auf das Grafikpaket und das Betriebssystem des verwendeten Rechners. Die meisten Komponenten von «Visual Prolog» sind in dieser Sprache geschrieben.