

VIP – DBS

EIN INTEGRIERTES LOGIK-ORIENTIERTES DATENBANK MANAGEMENT SYSTEM UNTER VIP-PROLOG

Thom Frühwirth, Eva Kühn
Institut für Praktische Informatik, Abteilung für Programmiersprachen und Übersetzerbau
Technische Universität Wien
A-1040 Wien

1. EINLEITUNG

An die Speicherung und Abfrage von Datenmengen werden immer größere Ansprüche gestellt. Wurde vor einem Jahrzehnt noch ein einfaches Datenbanksystem als ultima ratio betrachtet, so sind heutzutage intelligente Datenbanksysteme und sogenannte Expertensysteme in aller Munde. Vieles ist jedoch noch Theorie, denn existierende Datenbanksysteme weisen Mängel bezüglich Laufzeiteffizienz, Flexibilität und Benutzerfreundlichkeit auf:

1. Oft wäre es wünschenswert, unterschiedliche Datenbanksprachen zur Verfügung zu haben, um unterschiedlichen Benutzerklassen und unterschiedlichen Problemkategorien gerecht zu werden.

2. Es ist meist dem Benutzer oder dem Datenbankadministrator überlassen, effiziente Indexpfade über seinen Daten zu spezifizieren bzw. für eine geeignete Clusterung von Daten zu sorgen.

3. Die Portabilität von Datenbanksystemen auf verschiedene Computersysteme ist meist beschränkt.

4. Eine intelligente Datenbank soll es dem Benutzer ermöglichen, komplexe Integritätsbedingungen über Daten einfach und stimmig definieren zu können.

Die logikorientierte Programmiersprache PROLOG (PROgramming in LOGic) bietet für viele Problemklassen eine grundlegende Vereinfachung der Problemlösung. Speziell auf dem Gebiet der Datenbanksysteme wird PROLOG auf Grund seiner prädikatenlogisch bedingten Verwandtschaft zum Relationalen Datenbankmodell immer mehr an Bedeutung gewinnen ([APP/BEN 85], [LI 84], [G/M/N 85], [CL/MEL 81]).

2. VIP-PROLOG UND VIP-DBS

VIP (Vienna Integrated Prolog) wird am Institut für Praktische Informatik, Abteilung Programmiersprachen und Übersetzerbau, an der Technischen Universität Wien entwickelt. VIP ist eine PROLOG-Programmierungsumgebung, die ein Modulkonzept, einen

Windowmanager und effiziente Speicherverwaltung bietet. Ein intelligentes Debugging-System und ein PROLOG-orientierter Editor unterstützen die Programmentwicklung. Die wesentlichen Komponenten von VIP sind ein inkrementeller Compiler und ein Interpreter. PROLOG-Regeln und PROLOG-Fakten werden in den VIP-Zwischencode übersetzt, auf den der Interpreter auf Grund verschiedener Optimierungstechniken schnell zugreifen kann (siehe Abbildung 1 im Anhang).

VIP-DBS (VIP- Database System) ist ein Datenbanksystem, das innerhalb der VIP-Programmierungsumgebung entwickelt wurde ([FRU 86], [KUE/FRU 85], [VIP 85]). VIP-DBS ist als unabhängiger Systemmodul implementiert und kann auf jedem beliebigen PROLOG-System verwendet werden, welches das Konzept der Modularität unterstützt. VIP-DBS verwendet die Sprache PROLOG, um ein logik-orientiertes Datenbanksystem auf Basis des Relationenmodells zu entwickeln. PROLOG ist damit die einheitliche Entwurfs-, Implementierungs-, Datenrepräsentations- und Benutzersprache. Die Hauptforderung des Projektes ist es dabei gewesen, die Theorie relationaler Datenbanken stimmig auf den logischen Ansatz von PROLOG abzubilden, ohne PROLOG um vordefinierte Prädikate zu erweitern.

Alle Befehle auf der Command-Level-Schnittstelle, d.h. sämtliche Befehle zur Datenabfrage, -manipulation und -definition, können als PROLOG-Regeln und PROLOG-Fakten gesehen werden. Die Command-Level-Schnittstelle ist daher eine Ein-Tupel-Schnittstelle mit einer PROLOG-Syntax. Sie dient einerseits als Schnittstelle für einen High-Level-Applikationsgenerator, der zum Beispiel window-gesteuerte Ein-/Ausgabe vorsieht, andererseits als Schnittstelle für andere herkömmliche Datenabfragesprachen. So wurde etwa eine Schnittstelle für die weit verbreitete Datenbanksprache SQL implementiert, die es erlaubt, auf dieselben PROLOG-Daten via SQL zuzugreifen ([ORACLE 83], [LI 84], [BCS 81]). VIP-DBS wurde als offenes System entworfen, in dem auch der hierarchische und der Netzwerk-Ansatz verwirklicht werden können.

VIP-PROLOG bietet sehr gute Voraussetzungen zu einer solchen engen Kopplung zwischen der logischen Programmiersprache PROLOG und einem Relationalen Datenbanksystem. VIP-PROLOG kann beliebige Teile seiner internen Datenbank auf einem Sekundärspeicher abspeichern und unterstützt durch seine Modularität die Strukturierung der Daten. Da mehrere Module – bedingt durch einen dynamischen Swapmechanismus – gleichzeitig im Arbeitsspeicher gehalten werden können, ist eine hohe Zugriffseffizienz gerade bei Operationen mit mehreren beteiligten Relationen (vor allem Verbund) gegeben.

Interpretation von PROLOG-Klausen als Relationen

Wir stellen ein Tupel einer Relation in PROLOG als Faktum dar. Dabei entspricht jede Komponente des Tupels genau einem Argument des Faktums. Der Funktornamen des Faktums ist der Relationenname der zum Tupel gehörigen Relation. Betrachten wir folgendes Beispiel:

angestellter(maier, Hobby):- Hobby = 'Fischen'; Hobby = 'Faulenzen'.

Wenn wir nun eine Abfrage ?-angestellter(maier,Hobby) in PROLOG formulieren, so erhalten wir nacheinander folgendeTupel als Antwort: