Simone Rehm

Komplexe Objekte und Anfragen in der Programmierung

Simone Rehm

Komplexe Objekte und Anfragen in der Programmierung

Zur Integration von Datenbanken und Programmiersprachen

DUV Deutscher Universitäts Verlag
GABLER · VIEWEG · WESTDEUTSCHER VERLAG

Die Deutsche Bibliothek — CIP-Einheitsaufnahme

Rehm, Simone:
Komplexe Objekte und Anfragen in der Programmierung : zur Integration von Datenbanken und Programmiersprachen / Simone Rehm. — Wiesbaden : Dt. Univ.-Verl., 1993
(DUV : Informatik)
Zugl.: Karlsruhe, Univ., Diss., 1992

ISBN-13: 978-3-8244-2040-7 e-ISBN-13: 978-3-322-85803-0
DOI: 10.1007/978-3-322-85803-0

Der Deutsche Universitäts-Verlag ist ein Unternehmen der Verlagsgruppe Bertelsmann International.

Gedruckt auf chlorarm gebleichtem und säurefreiem Papier

Danksagung

Das vorliegende Buch stellt die von mir verfaßte und von der Fakultät für Informatik der Universität Karlsruhe genehmigte Dissertation dar. Sie entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin der Abteilung Programmstrukturen im Forschungszentrum Informatik (FZI) in Karlsruhe. In der datenbankspezifischen Ausrichtung des Themas spiegelt sich die über viele Jahre anhaltende, fruchtbare Zusammenarbeit mit Mitarbeiterinnen und Mitarbeitern der Abteilung Datenbanksysteme wider. Die in dieser Zeit gemeinsam bearbeiteten Projekte DAMOKLES und OBST boten ein geeignetes Forschungsumfeld für die in der Arbeit behandelte Frage nach der Integration von Anfragen in anweisungsorientierte Programmiersprachen.

Mein besonderer Dank gilt meinem Betreuer Herrn Prof. Goos, der mir einerseits den nötigen Freiraum für die Arbeit an der Dissertation gewährte und der andererseits dem Fortschreiten der Arbeit durch kritische Fragen und seine unermüdliche Diskussionsbereitschaft immer wieder wichtige Impulse verlieh. Ganz herzlich bedanken möchte ich mich auch bei Herrn Prof. Lockemann, der mich in der Abteilung Datenbanksysteme stets willkommen hieß und von dem ich als Zweitgutachter wesentliche, fachliche Anregungen erhielt.

Fachliche Unterstützung fand ich außerdem bei zahlreichen Kollegen und Studenten, denen ich an dieser Stelle ebenfalls danken möchte. Namentlich erwähnt seien Bernhard Schiefer und Franz Weber, deren Diskussionsbeiträge für mich sehr wertvoll waren. Dirk Ecker danke ich für seinen Einsatz bei der Implementierung der Eiffel-Klassen. Nicht selten gibt es während einer Dissertation auch Phasen der Stagnation, in denen Mut und Motivation nachlassen. Daß derartigen Tiefpunkten auch immer wieder ein Hoch folgte, ist vor allem das Verdienst meines Zimmerkollegen und Freunds Michael Ranft. Für die vielen kurzweiligen Stunden innerhalb und außerhalb des FZIs danke ich ihm. Dank gebührt auch Werner Rehm und Ulrich Hägele, die, obwohl fachfremd, doch erhebliche Mühe darauf verwendet haben, die Arbeit auf schriftliche und sprachliche Mängel hin durchzusehen.

Karlsruhe, im Dezember 1992

Simone Rehm

Vorwort

Während sich Programmiersprachen vorwiegend zur Formulierung von Algorithmen eignen, bieten Datenbanksysteme wertvolle Dienste beim Speichern und Wiederauffinden von Daten an. Bei der Programmierung von Datenbankanwendungen will man sich beide Vorteile zunutze machen, stößt dabei aber auf das Problem, daß von Seiten der Programmiersprache und von Seiten des Datenbanksystems unterschiedliche Modellierungskonzepte angeboten werden. So unterstützen Datenbanksysteme häufig die Modellierung strukturierter Datenmengen und die Formulierung deskriptiver Anfragen, während Programmiersprachen nach wie vor auf die Manipulation einzelner Datensätze ausgerichtet sind. Diese unterschiedlichen Konzepte führen im Fall der Kopplung von Programmiersprachen und Datenbanksystemen zwangsläufig zu Reibungsverlusten an der Schnittstelle zwischen beiden Systemen, die sich in aufwendigen und fehleranfälligen Typkonvertierungen äußern oder die Umformulierung deskriptiver Anfragen nach sich ziehen. Ziel der Integration von Datenbanken und Programmiersprachen ist es, solche Konvertierungen zu vermeiden. Letztlich will man zu einer Sprache gelangen, in der sich Datenbankobjekte von Programmobjekten nur noch in ihrer Lebensdauer, nicht jedoch in Art der Definition und Handhabung unterscheiden. Ein vielversprechender Weg in diese Richtung ist der Entwurf von Datenbankprogrammiersprachen, die aus der Erweiterung von Programmiersprachen um die Fähigkeiten eines spezifischen Datenmodells hervorgehen.

Betrachtet man jedoch existierende Datenbankprogrammiersprachen, so fällt auf, daß deskriptive Anfragen darin häufig kein adäquates Pendant finden. Oft genug muß die Iteration über den Suchraum noch explizit ausformuliert werden, oder es stehen nicht genügend aussagekräftige Sprachelemente zur Beschreibung von Suchbedingungen zur Verfügung. Dieses Buch befaßt sich daher systematisch mit dem Entwurf von Datenbankprogrammiersprachen, und zwar unter der Zielvorgabe, Anfragen zum integralen Bestandteil der Sprache werden zu lassen. Es untersucht vorweg, welche Voraussetzungen eine Programmiersprache erfüllen muß, um diese Zielvorgabe erfüllen zu können, und schildert dann im einzelnen die Maßnahmen der Integration. Auf Seiten der Programmiersprache erweisen sich Typkonzepte wie Datenabstraktion und die Definition parametrischer Typen als hilfreiche Stützen, während sich auf Seiten des Datenbanksystems *komplexe Objekte* als tragfähiges Gerüst für die Integration von Anfragen erweisen. Ihnen liegt als Konstruktionsprinzip die beliebige Kombinierbarkeit der Mengen- und der Tupelkonstruktion zugrunde. Zahlreiche Sachverhalte der Anwendungswelt können im Zuge der Datenmodellierung unmittelbar auf komplexe Objekte abgebildet werden, und in Anfragesprachen, die auf komplexen Objektmodellen basieren, sind vielseitige Anfragen formulierbar, auch solche, die das Auswahlvermögen des Relationenkalküls oder der relationalen Algebra übersteigen.

Ausgehend von diesen beiden Eckpfeilern wird zunächst konzeptionell ein Objektmodell geschaffen, das die Strukturierungsprinzipen der Mengen- und der Tupelkonstruktion und das Prinzip der Datenabstraktion in sich vereint. Dieses Objektmodell wird in ein flexibles, programmiersprachliches Typsystem eingebettet, das sich durch unterschiedliche Gleichheitsbegriffe für komplexe und abstrakte Typen und durch eine implizite Untertyprelation für komplexe Typen auszeichnet. Anhand einer Algebra werden Operatoren spezifiziert, mit denen sich komplexe Objekte inspizieren lassen und die den Grundstock für ein mächtiges Anfragekonzept bilden. Anschließend wird eine Teilsprache zur Definition und Manipulation komplexer Objekte präsentiert, in der sich diese Operatoren widerspiegeln. In Abhängigkeit von den in der Programmiersprache vorhandenen Typkonzepten wird schließlich gezeigt, wie diese Teilsprache mit einer vorgegebenen Zielsprache integriert werden kann. Somit wird der Übergang von einer Programmiersprache zu einer Datenbankprogrammiersprache praktisch nachvollziehbar gemacht. Dabei wird nachgewiesen,

- daß sich objektorientierte Sprachen besonders gut für die Integration von Anfragen eignen, da der stark ausgeprägte Polymorphismus dieser Sprachen es erlaubt, die erforderlichen Typprüfungen anschließend weitgehend im Rahmen der Zielsprache abzuhandeln;

- daß sich insbesondere parametrischer Polymorphismus in Kombination mit der Möglichkeit, die Typparameter entlang einer Typhierarchie einzuschränken, als wertvolles Hilfsmittel bei der Realisierung von Anfrageoperationen erweist;

- daß sich auch benutzerdefinierte Operationen auf einfache Weise in das Anfragekonzept miteinbeziehen lassen, falls die Zielsprache Funktionen höherer Ordnung unterstützt.

Am Beispiel der objektorientierten Programmiersprache Eiffel werden die Integrationsmaßnahmen konkretisiert. Im Anschluß an die Verankerung komplexer Objekte in einem programmiersprachlichen Typsystem kann die Integration mit einem Datenbanksystem so stattfinden, daß eine einheitliche Sprache entsteht, in der alle Objekte unabhängig von ihrem Typ persistent gemacht werden können und Anfragen gleichermaßen zur Inspektion temporärer wie persistenter Objekte zur Verfügung stehen.

Inhalt