Marco Pötke

Spatial Indexing for Object-Relational Databases

Herbert Utz Verlag · Wissenschaft
München

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Ein Titeldatensatz für diese Publikation ist
bei Der Deutschen Bibliothek erhältlich

Zugleich: Dissertation, München, Univ., 2001

ISBN 3-8316-0043-0

Printed in Germany

Herbert Utz Verlag GmbH, München
Tel.: 089/277791-00 – Fax: 089/277791-01

Table of Contents

Chapter 1
Introduction

1.1 Spatial Information Systems

The design and development of digital information systems is one of the most important areas in computer science. In order to store, manage, analyze, and present huge amounts of data, the required facts of the real world have to be extracted and decomposed into a set of manageable entities, e.g. *employees*, *departments*, and *revenues*. Data models are used to describe the objects and operations of the resulting *miniworld*. For many applications, one or more of the modeled entities represent extended objects in a one- or multidimensional space, including *contract periods* for personnel management, *real estates* for land registry, and *parts* for mechanical engineering. Spatial information systems are specialized to handle such spatial data properly.

1.1.1 Database Systems

A *database system* (DBS) represents the core component of any information system. It comprises a *database* (DB) and a *database management system* (DBMS) which stores, maintains, and retrieves data stored in the database. In contrast to a file-based organization, database systems offer substantial advantages for managing huge amounts of persistent data. Standard database services provide logical and physical data independence, transactions, concurrency control, integrity checking, recovery, security, standardization, and distribution [Dat 99]. Many data models have been

proposed for database systems, including hierarchical, network-based, relational, object-oriented, and object-relational approaches. In this thesis, we focus on the *object-relational data model*, as it is widely accepted and implemented by many database systems. Furthermore, its extensibility is a necessary precondition for the seamless embedding of spatial data types and operations.

1.1.2 Integrating Spatial Data

In order to design effective and efficient spatial information systems, specialized techniques for the *spatial data management* are required. Today, spatial data management has evolved from a stand-alone solution to a demanding database application, including geographic information systems [DeM 00], temporal databases [Sno 00], and computer-aided design [BKP 98]. The requirements of these application domains are challenging: the databases may contain millions of spatial and temporal extended objects. In addition, they are manipulated by thousands of concurrent users. To achieve interactive response times for spatial queries, spatial index structures [GG 98] [MTT 00] have to be provided. Unfortunately, only very few standard database systems offer built-in techniques for spatial indexing.

How can standard information systems be equipped with the appropriate techniques to cope with spatial data and queries? To what extent can these techniques exploit the robust infrastructure of existing database systems? Can off-the-shelf database kernels efficiently support spatial applications? In the present thesis, we will discuss these questions and propose solutions to supplement standard object-relational database systems with efficient spatial data management. These methods will be evaluated both, theoretically and practically.

1.2 Applications of Spatial Databases

In the following, we illustrate the impact of spatial information systems by outlining three fundamental applications: geographic information systems, mechanical engineering, and temporal databases. Instead of stand-alone spatial engines, they require the spatial data management to be closely connected to the management of non-spatial data. Such integrated solutions serve as vital foundations to build spatial data warehouses for on-line analytical processing, document management, and enterprise resource planning [Gün 99].

a) Geographic information systems

b) Mechanical engineering

Figure 1: Databases of multidimensional extended objects.

1.2.1 Geographic Information Systems

Geographic information systems (GIS) manage geo-referenced data, i.e. geographic phenomena associated with spatial regions of a planet surface [MP 94]. A GIS has to provide a combined analysis on spatial and non-spatial data. Figure 1a depicts an example for urban planning and development, a typical application of GISs. The spatial entities of the modeled miniworld include buildings, real estates, streets, railroads, parks, and rivers. Due to the huge amounts of data, spatial queries like “retrieve all buildings within a distance of 50 yards from the central park” have to be supported by specialized query processing methods and spatial index structures. There are many other important applications of geographic information systems, including agriculture, geology, environmental management, and spatial database marketing.

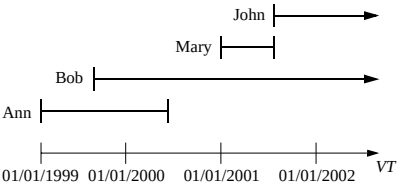
1.2.2 Mechanical Engineering

The underlying data space need not to be restricted to two dimensions. *Computer-aided design* (CAD) for mechanical engineering often requires three-dimensional data spaces. Figure 1b depicts some of the three million parts of a commercial airplane. Each part occupies an individual region in the product space. Spatial indexing is indispensable to process queries like “retrieve all parts in the spatial neighborhood of the jet engines” at interactive response times. Many CAD applications immediately benefit from an efficient and robust spatial data management, including the *digital mockup* (DMU) to check fit and appearance of product components and *haptic rendering* to simulate maintenance tasks in virtual environments.

employees

name	**contract**
John	07/01/2001 – *NOW*
Mary	01/01/2001 – 06/30/2001
Bob	08/15/1999 – *NOW*
Ann	01/01/1999 – 05/20/2000
…	…

a) Employees database



b) Valid time ranges

Figure 2: Database of one-dimensional extended objects.

1.2.3 Temporal Databases

Temporal databases record information which is varying with respect to the time. The validity of facts in the real world is modeled by intervals on a *valid time* dimension. Figure 2 shows a valid time database for a human resource department, where each employee is referenced to the duration of the corresponding contract. Intervals can be characterized as one-dimensional spatial objects, as they have a well-defined location and extension with respect to the time line. Spatial indexing can support queries like “find all employees as of 01/01/2000”. In addition to valid time, the *transaction time* indicates the period for which specific data is current in the database. Both time dimensions may be combined to two-dimensional domains supporting *bi-temporal databases*. Many database applications have a temporal nature, including scheduling applications such as airline reservations, financial applications such as portfolio management, and record-keeping applications such as medical information systems [JS 99].

1.3 Outline of the Thesis

The three examples given above already illustrate the importance of spatial data management for one- and multidimensional data spaces. This work discusses problems and solutions of spatial indexing for object-relational databases. It is divided into three major parts: *extensibility in database systems, relational access methods,* and *database integration for virtual engineering*.

1.3.1 Extensibility in Database Systems (Part I)

The first part introduces the basic characteristics of spatial data models in extensible database systems. Chapter 2 describes spatial data types and spatial predicates. Fundamental query primitives are presented, including spatial selection, spatial ranking, and spatial joins. The multi-step approach to evaluate spatial predicates is explained, and finally, three basic approaches to design a spatial database architecture are reported.

Chapter 3 discusses the possibilities and limitations of extensibility in object-relational database systems. For the seamless integration of spatial index structures, two basic interfaces have to be addressed: first, a high-level interface is required to embed index operations within the declarative query language of the target database system. Second, a low-level interface of the index structure to the database kernel has to be implemented, including access to the persistent storage manager, concurrency control, and recovery services. We will show that the high-level interface is well supported by common extensible indexing frameworks, whereas the low-level components of most existing database systems lack specific support for extensible indexing.

1.3.2 Relational Access Methods (Part II)

As a solution to the above problem, chapter 4 presents the concept of *relational access methods*. This class of index structures can be implemented on top of off-the-shelf database servers without any internal modifications or augmentations to existing kernels. Thereby, spatial and temporal applications can potentially benefit from the full functionality of the underlying database system, including transactions, high concurrency, and efficient recovery. We will develop the main properties of relational access methods, and illustrate the presented concepts by some examples.

Chapter 5 introduces the Relational Interval Tree, an efficient and scalable relational access method to manage one-dimensional interval data. We will show that its I/O complexity for updates and interval intersection queries is optimal for many application domains. Empirical evaluations on databases containing up to 1,000,000 intervals demonstrate that relational access methods are not only smart to implement and to employ, but also deliver a very high performance.

Chapter 6 widens the scope from one-dimensional to multidimensional extended objects. In a first step, the management of interval data is generalized to the storage and retrieval of arbitrary interval sequences which also occur in many temporal data-

base applications. In a second step, the concept of space-filling curves is applied to map multidimensional extended objects to interval sequences. In an empirical evaluation on databases ranging up to 7 GB in size, we compare the interval sequence approach with existing techniques based on regular tiling and box decomposition.

Chapter 7 completes the object-relational integration of spatial indexing by developing a cost model to estimate the selectivity, I/O cost, and CPU cost of one- and multidimensional queries on the Relational Interval Tree. The proposed functions can be attached to the object-relational optimizer by means of a high-level extensible indexing framework. Consequently, the declarative paradigm of relational query languages is preserved. An empirical evaluation on the accuracy of the presented techniques provides a proof of concept.

1.3.3 Database Integration for Virtual Engineering (Part III)

The third part of this thesis presents an in-depth case study of spatial indexing in object-relational databases. In cooperation with the Volkswagen AG, Wolfsburg, the German Aerospace Center DLR e.V., Oberpfaffenhofen, and the Boeing Company, Seattle, we have studied different applications in the area of mechanical engineering. Chapter 8 motivates the basic benefits of introducing database support for virtual engineering into the file-based world of computer-aided design (CAD). As we integrate 3D spatial data management into standard object-relational database systems, the required support for data independence, transactions, recovery, and interoperability can be achieved.

In chapter 9, we present some basic representations, transformations, and operations on three-dimensional engineering data. These techniques are vital contributions to spatial data management for CAD databases. Chapter 10 assembles the fundamental building blocks developed throughout this thesis to a complete system architecture for the database integration of virtual engineering (DIVE). By using relational storage structures, the DIVE system provides three-dimensional spatial data management within a commercial database system. The functionality and performance of DIVE is evaluated on real CAD data of a car project. Finally, chapter 11 recapitulates the main contributions of this thesis and suggests some directions for future work.