

Hermes - A Framework for Location-Based Data Management*

Nikos Pelekis, Yannis Theodoridis,
Spyros Vosinakis, and Themis Panayiotopoulos

Dept of Informatics, University of Piraeus, Greece
{npelekis, ytheod, spyrosv, themisp}@unipi.gr
<http://isl.cs.unipi.gr/db>

Abstract. The aim of this paper is to demonstrate Hermes, a robust framework capable of aiding a spatio-temporal database developer in modeling, constructing and querying a database with dynamic objects that change location, shape and size, either discretely or continuously in time. Hermes provides spatio-temporal functionality to state-of-the-art Object-Relational DBMS (ORDBMS). The prototype has been designed as an extension of STAU [6], which provides data management infrastructure for historical moving objects, so as to additionally support the demands of real time dynamic applications (e.g. Location-Based Services - LBS). The produced type system is packaged and provided as a data cartridge using the extensibility interface of Oracle10g. The offspring of the above framework extends PL/SQL with spatio-temporal semantics. The serviceableness of the resulting query language is demonstrated by realizing queries that have been proposed in [9] as a benchmarking framework for the evaluation of LBS.

1 Introduction

In recent years, we have been witnessing the explosion of emerging non-traditional database applications, such as location-based services. The main components of the underlying database of such applications include stationary and moving objects. The so-called Moving Objects Databases (MODs) are (or soon will be) ubiquitous. As the number of mobile commerce or, in general, mobile services, increases rapidly everyday, the need for robust management systems about location data, as well as the analysis of user movements are vital. In this paper, we present the design and implementation issues of a research prototype for efficient location-based data management, called Hermes (the ancient Greek god of Commerce). Hermes can be considered as a MOD management system with emphasis on the peculiarities of MODs, from representation to querying issues. Someone could mention a series of applications of Hermes at various levels in the context of mobile services. For example, Hermes can be used as a plug-in in telecom companies' data warehouses

* Research partially supported by the Pythagoras EPEAEK II Programme of the Greek Ministry of National Education and Religious Affairs, co-funded by the European Union.

that include spatio-temporal content. The previous example refers to offline processing of such historical data. Besides, Hermes supports the data management of real-time mobile services, addressing the issues of modern dynamic applications. For instance, imagine a user (tourist, consumer) moving around a city equipped with a next-generation mobile terminal (e.g. 3G cell-phone or PDA enhanced by the presence of a GPS receiver), receiving hints of information, commercial spots etc. Researchers [1], [2], [3], [4], motivated from such kind of application scenarios have tried to model spatio-temporal databases using this concept of moving objects and integrate them into any extensible DBMS. On the other hand, commercial relational or object-relational database systems offer limited capability of handling this kind of non-traditional data (object trajectories, in time and space). Hermes is the partial realization of the above discussed research vision.

2 The Prototype

Hermes is developed as a system extension that provides spatio-temporal functionality to Oracle10g's Object-Relational Database Management System (ORDBMS). The system is designed in a way that it can be used either as a pure temporal or a pure spatial system, but its main functionality is to support the modelling and querying of continuously moving objects. Such a collection of data types and their corresponding operations are defined, developed and provided as an Oracle data cartridge. *Hermes Moving Data Cartridge (Hermes-MDC)* is the core component of the Hermes system architecture. *Hermes-MDC* provides the functionality to construct a set of moving, expanding and/or shrinking geometries, as well as time-varying base types. Each one of these moving objects is supplied with a set of methods that facilitate the cartridge user to query and analyze spatio-temporal data. Embedding this functionality offered by *Hermes-MDC* in Oracle's DML [5], one obtains an expressive and easy to use query language for moving objects.

In order to implement such a framework in the form of a data cartridge we exploit a set of standard data types together with the static spatial data types offered by the *Spatial* option of Oracle10g [5] and the temporal literal types introduced in a temporal data cartridge, called *TAU Temporal Literal Library Data Cartridge (TAU-TLL)* [6]. Based on these temporal and spatial object data types *Hermes-MDC* defines a series of moving object data types illustrated in the UML class diagram of Figure 1. The interested reader in a detailed discussion for the resulted type system is referred to [7].

3 Architecture of Hermes

Figure 2 illustrates the architecture of the Hermes system. A straightforward utilization scenario for a *Hermes-MDC* user is to design and construct a spatio-temporal object-relational database schema and build an application by transacting with this database. In this case, where the underlying ORDBMS is Oracle10g, in order to specify the database schema, the database designer writes scripts in the syntax of the *Data Definition Language (DDL)*, which is the PL/SQL, extended with the spatio-temporal operations previously mentioned.

To build an application on top of such a database for creating objects, querying data and manipulating information, the application developer writes a source program in *Java* wherein he/she can embed *PL/SQL scripts* that invoke object constructors and methods from Hermes-MDC. The *JDBC pre-processor* integrates the power of the programming language with the database functionality offered by the extended *PL/SQL* and together with the *ORDBMS Runtime Library* generate the application's executable. By writing independent stored procedures that take advantage of Hermes functionality and by compiling them with the *PL/SQL Compiler*, is another way to build a spatio-temporal application.

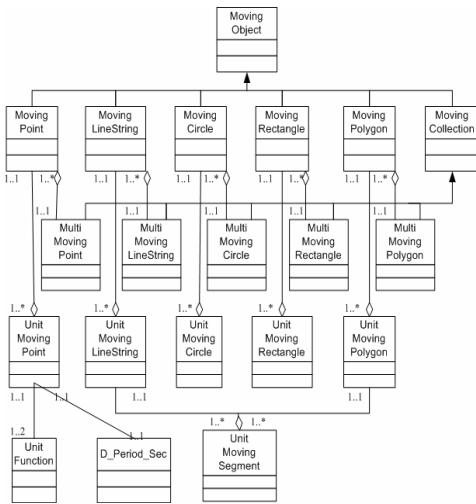


Fig. 1. Hermes-MDC Class Diagram

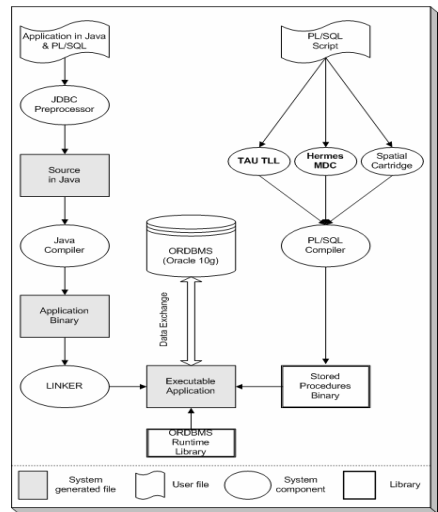


Fig. 2. The architecture of Hermes

4 The Demonstration Scenario

In order to demonstrate the usefulness and applicability of the server-side extensions provided by Hermes we implement the majority of the benchmark queries for LBS proposed in [9]. Additionally, we specially develop an LBS application scenario for travellers entering the area of an airport, construct a spatial database modeling the ground plan of the airport, and input random trajectories of travellers moving around the area. Then, we pose queries following the same classification as proposed in [9]. Indicative examples include:

- *Queries on stationary reference objects*; examples include point (e.g. does this check-in serve my flight?), range (e.g. are there any fellow travellers in the area in front of this check-in?), distance-based (e.g. find the closer check-in), nearest-neighbor (e.g. find the closest coffee shops to my current location) and topological queries (e.g. find travellers crossed this gate during the past hour);

- *Queries on moving reference objects*; examples include distance-based (e.g. find travellers passed close to me this evening) and similarity-based queries (e.g. find the three most similar trajectories to the one I have followed so far in the airport);
- *Join queries*; examples include distance-join (find the closest check-ins to travellers of this flight) and similarity-join queries (find the two most similar pairs of travellers' trajectories);
- *Queries involving unary operators*, such as travelled distance or speed (e.g. find the average speed of travellers on Saturday nights).

Based on related research work [8] the above queries constitute a minimum functionality a MOD system should provide. The above demonstration scenario is also accompanied (wherever appropriate) by visual illustrations formulated by MapViewer [5] (see Figure 3).

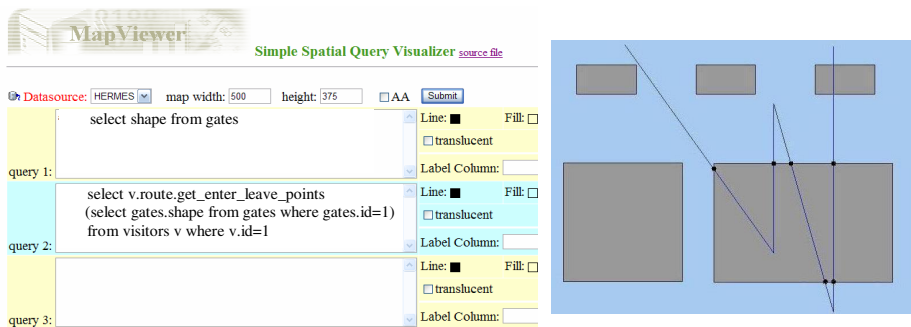


Fig. 3. Visualization of entry and exit points in an area of interest

5 Conclusions

In this paper, the Hermes prototype system was introduced. Hermes has been designed as a system extension that provides spatio-temporal functionality to state-of-the-art ORDBMS. The usability of the prototype is demonstrated by applying benchmark queries proposed for the evaluation of systems providing location based services. Future work includes knowledge discovery from MODs. Typical examples we plan to incorporate in Hermes framework are: ‘characterization of routes as (in)frequent’, ‘habits in motion with respect to space and/or time constraints’ etc.

References

1. S. Dieker and R. H. Guting, “Plug and Play with Query Algebras: Secondo. A Generic DBMS Development Environment”, In *Proc. of the Int. Database Engineering and Applications Symposium (IDEAS)*, p. 380-390, 2000.
2. M. Erwig, R.H. Guting, M. Schneider, and M. Vazirgiannis, “Spatio-Temporal Data Types: An Approach to Modelling and Querying Moving Objects in Databases”, *GeoInformatica*, 3(3):265-291, 1999.

3. L. Forlizzi, R. H. Guting, E. Nardelli, M. Schneider, "A Data Model and Data Structures for Moving Objects Databases", In *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pages 319-330, 2000.
4. R.H. Guting, M. H. Bohlen, M. Erwig, C. S. Jensen, N. A. Lorentzos, M. Schneider, and M. Vazirgiannis, "A Foundation for Representing and Querying Moving Objects", *ACM Transactions on Database Systems*, 25(1):1-42, 2000.
5. Oracle Corp. *Oracle®*. Oracle Database Documentation Library, 10g Release 1 (10.1), URL: <http://otn.oracle.com/pls/db10g/>.
6. N. Pelekis. STAU: A spatio-temporal extension to ORACLE DBMS, PhD Thesis, UMIST, 2002.
7. N. Pelekis, B. Theodoulidis, Y. Theodoridis and I. Kopanakis. An Oracle Data Cartridge for Moving Objects. UNIPi-ISL-TR-2005-01, March 2005. URL: <http://isl.cs.unipi.gr/db>.
8. N. Pelekis, B. Theodoulidis, I. Kopanakis and Y. Theodoridis, "Literature Review of Spatio-Temporal Database Models", *Knowledge Engineering Review*, 19(3):235-274, 2005.
9. Y. Theodoridis. "Ten Benchmark Database Queries for Location-based Services", *The Computer Journal*, 46(6):713-725, 2003.