Spatio-Temporal Databases in Practice: Directly Supporting Previously Developed Land Data using Tripod

Tony Griffiths, Alvaro A.A. Fernandes, Norman W. Paton,
Seung-Hyun Jeong and Nassima Djafri
Department of Computer Science
University of Manchester, Manchester M13 9PL, UK
{griffitt|alvaro|norm|jeong|ndjafri@cs.man.ac.uk}

Keith T. Mason
School of Earth Sciences and Geography
University of Keele, Staffordshire ST5 5BG, UK
k.t.mason@esci.keele.ac.uk

1 Introduction

This demo presents a complete spatio-temporal object DBMS called *Tripod* [2], in the context of an application to support the management of previously developed land (PDL). In particular the demo focuses on: the techniques that are available to realise the application data model; the facilities necessary to realise the operational semantics of updates, and the facilities that are available to identify interesting patterns of spatio-temporal change.

Whilst there exist other proposals for data models and query languages for spatio-temporal object databases, Tripod is unique in that it provides: a complete implementation of an expressive spatio-temporal data model to allow modelling of (a)spatial time-varying data; the first example of an implementation of language bindings to support manipulation of stored data; and the first complete implementation of a spatio-temporal OQL that we know of.

2 Case Study

The demo focuses on the benefits arising from use of a spatio-temporal DBMS on an important category of application, namely land parcel management. The most recent initiative in this field, in the UK, is the NLUD (National Land Use Database) project (http://www.nlud.org.uk/). It aims to provide an up-to-date and detailed geographical record of land use in England.

Within NLUD, an individual PDL parcel can be, at any point in time, in one of five thematic states. This typology is based upon a cycle of development that recognises that sites have ceased to be actively used, become available for redevelopment, and subsequently return to regular use.

In many cases, such changes are straightforward to model, and present few problems for the management of site histories. However, there are likely to be many cases where the progress from one state to another is much less straightforward. In particular, there will be cases where thematic change only applies to part of the site. The problem here is that such change triggers changes to parcel configurations: their shapes, sizes, and topological relationships with other parcels. In such cases, it is the geometric processes of fusion, fission or reallocation that must be catered for in tracking the life histories of individual sites. In the absence of appropriate database infrastructure, the detail of such changes can only be determined from visual inspection of each 'snapshot'.

Several systems currently exist that manage land use for various government agencies [1]. Each of these provides a bespoke solution, utilising its own particular data model and suffering from the lack of integrated database support for (a)spatial historical storage. This lack of support causes developers problems throughout the development life-cycle, as update programs become overly complex, and their querying mechanisms have difficulty in exploiting any opportunities for optimisation.

3 The Tripod System

Tripod extends the ODMG standard for object databases [2] so that past states of all ODMG types, including spatial types, can be recorded using a specialised mechanism called a *history*. Tripod provides two interfaces to data; a declarative query interface and an imperative programming interface. The query interface is a spatio-historical extension to OQL, whereas the programming interface maps object model constructs into programming language constructs within an existing programming language.

The main components in the Tripod architecture are: a persistent store that loads and saves objects to and from the database and maintains metadata; a query processor that optimises and executes OQL queries; and a programming language binding that provides imperative language access

to the database.

4 Using Tripod for PDL

The definition of a Tripod database consists of two parts: a schema (defined using a declarative ODL) to specify the structure of user-defined types and their behaviour, and an implementation of each of these behaviours specified using a programming language binding. The ODL preprocessor both creates and initialises persistent data structures and automatically generates header files that allow programs to manipulate persistent objects. Once the application program and type information is compiled into object code, it is linked with libraries that implement the Tripod runtime system and the persistent store. The output of this process is an executable application that interacts with the underlying ODBMS. When the state of a database needs to be queried, developers can either write native language application programs or issue declarative OQL queries. Tripod's OQL extends that defined by the ODMG with spatial and spatio-historical constructs.

4.1 Data Modelling

The demo illustrates how Tripod histories automatically support the maintenance of relationships between time-varying database entities such as land parcels, their administrative regions and owners, and how Tripod's history mechanism is a much more general and powerful mechanism than the methods utilised by bespoke systems.

A simplified version of the PDL schema is shown in Figure 1. Note that the object type PDL site is historical, whereas Owner is not, indicating that only the history of the relationships between a PDL site and its owner are of interest, rather than the history of owners. This schema also illustrates Tripod's orthogonal treatment of spatial (e.g., extent) and aspatial properties (e.g., land type).

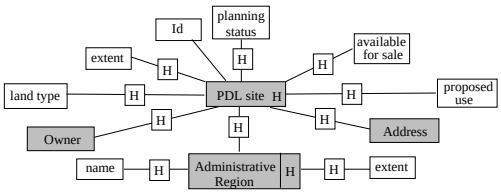

Figure 1. Tripod PDL Schema

4.2 Manipulating Data

The second phase of the demo develops an application to populate a database, illustrating how developers utilise the underlying Tripod model and C++ libraries to manipulate stored data. To the best of our knowledge, this is the first implementation of a programmatic interface to support population and maintenance of spatio-temporal data.

Finally, we illustrate how Tripod supports the maintenance of historical relationships between objects. For example, if a land parcel is deleted from the database for part of its valid time, Tripod provides developers with facilities to automatically update the valid time of any historical relationships that the object participated in. The exact semantics of this cascading update are parameterisable by the application programmer, but require no developer effort.

4.3 Analysing Data

This phase of the demo shows how developers can use either Tripod's language bindings or its spatio-temporal OQL to query the state of a PDL database. The Tripod OQL evaluator makes use of specialist algorithms, e.g., joins and aggregation. These are made available to the programmer through the `d_OQL_Query` class. E.g.: *Which sites bordered site 460500382 ?*

```
d_Bag<d_Ref<PDL_site> > someSites;
d_OQL_Query q1("SELECT p2.Id FROM p1 in PDL_sites,
  p2 in PDL_sites, gext1 in p1.extent,
  gext2 in p2.extent WHERE p1.Id = $1 and
  gext1.value.border_in_common(gext2.value) and
  gext1.validTime.common_points(gext2.validTime)
  and p1 != p2");
q1 << "460500382";  d_oql_execute(q1, someSites);
d_Ref<PDL_site> aSite; // iterate over results of q1
d_Iterator<d_Ref<PDL_site> > it = \
            someSites.create_iterator();
while(it.next(aSite)) {
  cout << "Site: " << aSite->get_Id() << endl;
  cout << aSite->get_extent().CurrentValue().area();
}
```

Acknowledgements. Discussions with Mike Worboys, John Stell, Chris Johnson and Bob Barr have contributed to shape the contributions of Tripod. Tripod is funded by the UK Engineering and Physical Sciences Research Council (EPSRC) and the Paradigm EPSRC platform grant.

Tripod is available for download to interested researchers from `http://www.cs.manchester.ac.uk/tripod/`.

References

[1] Ian N. Gregory. Time-variant gis databases of changing historical administrative boundaries: A european comparison. *Transactions in GIS*, 6(2):161–178, 2002.

[2] Tony Griffiths, Alvaro A. A. Fernandes, Norman W. Paton, and Robert Barr. The tripod spatio-historical data model. *Data Knowl. Eng.*, 49(1):23–65, 2004.