

# Management of Dynamic Location Information in DOMINO

O. Wolfson<sup>1,2,\*</sup>, H. Cao<sup>1</sup>, H. Lin<sup>1</sup>,  
G. Trajcevski<sup>1</sup>, F. Zhang<sup>1</sup>, and N. Rishe<sup>3</sup>

<sup>1</sup> Department of Computer Science, University of Illinois,  
Chicago, IL 60607

<sup>2</sup> Mobitrac, Inc.,

Chicago, IL 60610, {wolfson,hcao,hlin,gtrajcev,fzhang}@cs.uic.edu

<sup>3</sup> Florida International Univ., School of CS,  
University Park, Miami, FL 33199

## 1 Background

Consider a database that represents information about moving objects and their location. For example, for a database representing the location of taxi-cabs a typical query may be: *retrieve the free cabs that are currently within 1 mile of 33 N. Michigan Ave., Chicago* (to pick-up a customer); or for a trucking company database a typical query may be: *retrieve the trucks that are currently within 1 mile of truck ABT312* (which needs assistance); or for a database representing the current location of objects in a battlefield a typical query may be: *retrieve the friendly helicopters that are in a given region*, or, *retrieve the friendly helicopters that are expected to enter the region within the next 10 minutes*. The queries may originate from the moving objects, or from stationary users. We will refer to applications with the above characteristics as moving-objects-database (MOD) applications, and to queries as the ones mentioned above as MOD queries. In the military MOD applications arise in the context of the digital battlefield (c.f [1]), and in the civilian industry they arise in transportation systems. For example, Omnitrac developed by Qualcomm (see [3]) is a commercial system used by the transportation industry, which enables MOD functionality. It provides location management by connecting vehicles (e.g. trucks), via satellites, to company databases. The vehicles are equipped with a Global Positioning System (GPS), and they automatically and periodically report their location.

Tracking using a moving-objects-database also enables futuristic applications such as augmented reality, where various images, charts, and other voluminous data (that cannot be stored in a portable/wearable computer for a large geographic area) is delivered “just-in-time” to the mobile computer. The delivered information pertains only to the geographic location in the immediate vicinity of the mobile computer, which continuously changes. In electronic commerce, tracking enables delivery of location-dependent dynamic travel information (e.g. local traffic conditions, local sales of interest) to a mobile subscriber.

---

\* Research supported by ARL Grant DAAL01-96-2-0003, NSF Grants ITR – 0086144, CCR – 9816633, CCR – 9803974, IRI – 9712967, EIA – 000516, INT – 9812325

## 2 The Demonstration

Our DOMINO prototype is intended to serve as a platform, or a toolkit for developing MOD type of applications. The system is the third in a three-layer architecture. The first layer is an Object Relational DBMS. The database stores the information about each moving object, including its plan of motion. The second layer is a GIS that adds capabilities and user interface primitives for storing, querying, and manipulating geographic information. The third layer, DOMINO, adds temporal capabilities, capabilities of managing the uncertainty that is inherent in future motion plans, capabilities for location prediction, and a simulation testbed. Currently, DOMINO runs on both Unix and MS/Windows. On both platforms DOMINO uses the Arc-View GIS. It uses the Informix DBMS on Unix, and Oracle on MS/Windows.

We will demonstrate the following features of DOMINO.

### 2.1 Location Modeling

The database may have various levels of information about the location of a moving object. It may know the current exact point-location, or it may know a general area in which the object is located but not the exact location, or it may know an approximate motion plan (e.g. traveling north on I95, at 60 miles per hour), or it may know the complete motion plan. The motion plan of a moving object is a sequence of way time points,  $(p_1, t_1), (p_2, t_2), \dots (p_n, t_n)$ , indicating that the unit will be at geographic point  $p_1$  at time  $t_1$ , at geographic point  $p_2$  (closer to the destination than  $p_1$ ) at time  $t_2$  (later than  $t_1$ ), etc. DOMINO supports all these levels of location information. In order to do so efficiently, it employs the concept of a dynamic attribute, i.e. an attribute whose value changes continuously as time progresses, without being explicitly updated. So, the location of a moving object is given by its dynamic attribute, which is instantiated by the motion plan of the object.

In DOMINO a motion plan is specified interactively by the user on a GIS on a map. DOMINO is currently using maps from GDT Corp. ([2]); the map contains the length of each city block, the coordinates of its endpoints, and the average traffic speed along each city block. The speed information in the GDT maps is static, but we update it using real-time traffic information collected periodically from a web site (<http://www.ai.eecs.uic.edu/GCM/CongestionMap.html>). Based on this information the current location of the object is computed at any point in time. This motion plan may be automatically updated by GPS information transmitted from the moving object.

In DOMINO a moving object can use one of several policies to update the locations database. One of the policies uses a cost based approach approach to quantify the tradeoff between uncertainty and communication cost. Specifically, a moving object updates the database whenever the deviation from the database location exceeds a prespecified bound  $b$  given in terms of distance or time [5]. The update includes a revised plan and possibly a new bound on the deviation. The bound  $b$  is computed using on a cost based approach that takes into consideration

the cost of update messages (in terms of wireless bandwidth consumption and processing power) and the cost of the deviation.

We collected updates by driving several trajectories, about 40 miles each, in the Chicago metropolitan area.

## 2.2 Spatio-temporal Capabilities

Maintaining motion plan information enables the system to answer queries pertaining to the current, future or past locations of the moving object, for example: *Q1 = Retrieve the moving objects that are expected to intersect a region R sometime during a given time interval I.* (I may be a time interval that lies entirely in the future, i.e. after the time when *Q1* is entered).

We will demonstrate the spatial and temporal primitives of the query language and its answer-display screen. The primitives are given in graphical format, and they can be combined with textual SQL in a natural and intuitive way. For example, in the query *Q1* above the region *R* may be drawn with a mouse on a real GIS map, and the time interval *I* may be specified on a graphical timeline. Information about the moving objects that satisfy the query is displayed in textual form, and the location of each such moving object is displayed as a point on the map. The spatio-temporal functions constitute a data-blade in an ORDBMS.

## 2.3 Uncertainty

We will demonstrate the capabilities of the query language and its answer-display screen in dealing with uncertainty. These include POSSIBLY and DEFINITELY semantics for queries. In other words, the query *Q1* above can be specified with POSSIBLY or DEFINITELY semantics [4]. Under the DEFINITELY semantics, an object will be retrieved if all of its possible trajectories intersect the region *R* during the interval *I*. Under the POSSIBLY semantics, an object will be retrieved if some of its possible trajectories intersect the region *R* during the interval *I*.

## References

1. S. Chamberlain. Model-based battle command: A paradigm whose time has come. In *Symposium on C2 Research and Technology*, 1995.
2. Geographic Data Technology Co. *GDT Maps*, 2000. <http://www.geographic.com>.
3. Qualcomm. *OmniTRACS, Communicating Without Limits*. <http://www.qualcomm.com>.
4. G. Trajcevski, O. Wolfson, F. Zhang, and S. Chamberlain. The geometry of uncertainty in moving objects databases. In *EDBT*, 2002. (to appear).
5. O. Wolfson, A. P. Sistla, S. Chamberlain, and Y. Yesha. Updating and querying databases that track mobile units. *Distributed and Parallel Databases*, 7, 1999.
6. O. Wolfson, B. Xu, S. Chamberlain, L. Jiang. Moving Objects Databases: Issues and Solutions. In *SSDB*, 1999.