02-MD

Management of Dynamic Location Information in DOMINO

O. Wolfson[1,2,*], H. Cao[1], H. Lin[1],
G. Trajcevski[1], F. Zhang[1], and N. Rishe[3]

[1] Department of Computer Science, University of Illinois,
Chicago, IL 60607
[2] Mobitrac, Inc.,
Chicago, Il 60610, {wolfson,hcao,hlin,gtrajcev,fzhang}@cs.uic.edu
[3] 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, Omnitracs 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

C.S. Jensen et al. (Eds.): EDBT 2002, LNCS 2287, pp. 769–771, 2002.

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, (p1,t1), (p2,t2),... (pn,tn), indicating that the unit will be at geographic point p1 at time t1, at geographic point p2 (closer to the destination than p1) at time t2 (later than t1), 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 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.

02-MD

Management of Dynamic Location Information in DOMINO

O. Wolfson[1,2,*], H. Cao[1], H. Lin[1],
G. Trajcevski[1], F. Zhang[1], and N. Rishe[3]

[1] Department of Computer Science, University of Illinois,
Chicago, IL 60607
[2] Mobitrac, Inc.,
Chicago, Il 60610, {wolfson,hcao,hlin,gtrajcev,fzhang}@cs.uic.edu
[3] 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, Omnitracs 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

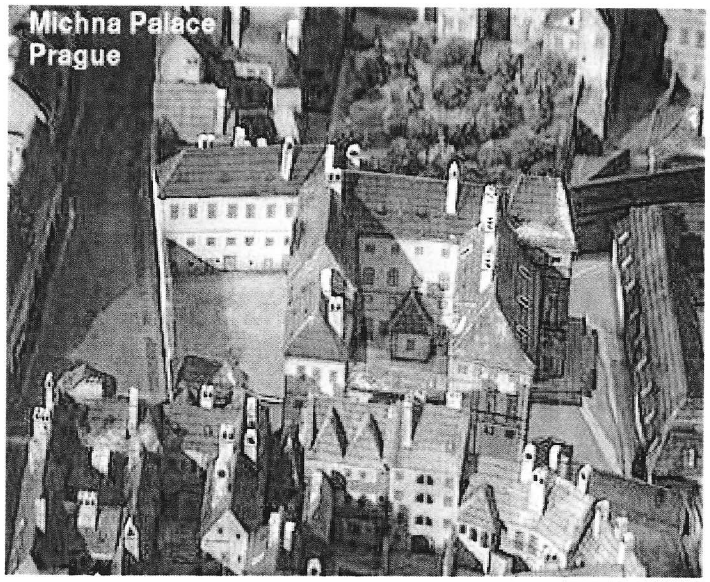
C.S. Jensen et al. (Eds.): EDBT 2002, LNCS 2287, pp. 769–771, 2002.

EDBT 2002

VIII. Conference on Extending Database Technology

March 24-28 2002, Prague

Michna Palace - part of Tyrs House



ENTER

Management of Dynamic Location Information in DOMINO

O. Wolfson[1,2]* H. Cao[1] H. Lin[1] G. Trajcevski[1] F. Zhang[1] N. Rishe[3]

[1] Department of Computer Science, University of Illinois, Chicago, IL 60607
[2] Mobitrac, Inc., Chicago, Il 60610, {wolfson, hcao, hlin, gtrajcev, fzhang}@cs.uic.edu
[3] 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, Omnitracs 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, (p1,t1), (p2,t2),... (pn,tn), indicating that the unit will be at geographic point p1 at time t1, at geographic point p2 (closer to the destination than p1) at time t2 (later than t1), 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 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.

This article was processed using the LaTeX macro package with LLNCS style

EDBT 2002

Promoted by the
EDBT Endowment

I n cooperation with the
VLDB Endowment
VLDB 2002

Sponsors & Supporting Organizations

This site is hosted by
Charles University,
Faculty of Math and Physics,
School of Computer Science

EDBT 2002

VIII. Conference on Extending Database Technology

March 24-28 2002, Prague Michna Palace - part of Tyrs House

EDBT has established a prestigious tradition by organizing a international conference on databases, in Europe, every two conference provides a unique opportunity for database resea practitioners, developers, and users to explore new ideas an tools, techniques, and experiences. The previous events wer Venice (1988 and 1990), Vienna (1992), Cambridge (1994), (1996), Valencia (1998), and Konstanz (2000).

Conference proceedings will be published by Springer-Ve LNCS series.

02-MD ~~02-MD~~
Page # 769 - 771
Wolfson-Eal-02-MD

Management of Dynamic Location Information in DOMINO

O. Wolfson[1 2]* H. Cao[1] H. Lin[1] G. Trajcevski[1] F. Zhang[1] N. Rishe[3]

[1] Department of Computer Science, University of Illinois, Chicago, IL 60607
[2] Mobitrac, Inc., Chicago, Il 60610, {wolfson, hcao, hlin, gtrajcev, fzhang}@cs.uic.edu
[3] 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, Omnitracs 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, (p1,t1), (p2,t2),... (pn,tn), indicating that the unit will be at geographic point p1 at time t1, at geographic point p2 (closer to the destination than p1) at time t2 (later than t1), 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 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.

mb

Call Number: QA76 L4

Location: General

Maxcost: $30.00

ILL: 8962420

DateReq: 8/7/2002

Date Rec: 8/9/2002

Borrower: FXG

LenderString: DLM,*IWA,IAC,IAH,JNA

Yes

No

Conditional

Title: Lecture notes in computer science.

Author:

Edition:

Imprint: Berlin ; New York : Springer-Verlag,

Article: Wolfson, O./Management of Dynamic Location Information in DOMINO

Vol:

No.: 2287

Pages: 769-772

Date: March 24-28, 2002

Borrowing Notes:

Fax:

```
ILL: 8962420      :Borrower: FXG      :ReqDate: 20020807 :NeedBefore: 20020906
:Status: IN PROCESS 20020809          :RecDate:          :RenewalReq:
:OCLC: 3719235    :Source: OCLCILL    :DueDate:          :NewDueDate:
:Lender: DLM,*IWA,IAC,IAH,JNA
:CALLNO:    :TITLE:   Lecture notes in computer science.   :IMPRINT: Berlin ; New
York : Springer-Verlag,   :ARTICLE: Wolfson, O./Management of Dynamic Location
Information in DOMINO   :VOL:       :NO: 2287   :DATE: March 24-28, 2002   :PAGES:
769-772          :VERIFIED: OCLC ISSN: 0302-9743 [Format: Serial]   :PATRON:  RISHE
NAPHTALI/50451   :SHIP TO:  Interlibrary Loan, GL-273 / Florida International
University /  Tamiami Trail / Miami, FL 33199 USA   :BILL TO:  (see above) Your
Fein# REQUIRED on bills. FXG's FEIN#59-6001874   :SHIP VIA: Library Rate or DLLI
(FLIN only)  :MAXCOST: $30.00    :COPYRT COMPLIANCE: CCG   :FAX: (305)348-6055,
ARIEL 131.94.104.95   :E-MAIL: upill@fiu.edu   :BILLING NOTES: ILL Order numbers
REQUIRED on invoices for processing!   :AFFILIATION:
SEFLIN,FLIN/DLLI,SOLINE/SO6,ASERL,LVIS,various reciprocals   :LENDING
CHARGES:                :SHIPPED:          :SHIP INSURANCE:          :LENDING
RESTRICTIONS:    :LENDING NOTES:    :RETURN TO:    :RETURN VIA:
```

ShipVia: Library Rate or

NeedBy: 9/6/2002

Return To:

Interlibrary Loan
Room 184 Parks Library
Iowa State University
Ames, IA 50011-2140

Ship To:

Interlibrary Loan, GL-273
Florida International University
Tamiami Trail
Miami, FL 33199 USA

ILL: **8962420** Borrower: FXG

Req Date: **8/7/2002** OCLC #: 3719235

Patron: **RISHE NAPHTALI/50451**

Author:

Title: **Lecture notes in computer science.**

Article: Wolfson, O./Management of Dynamic Location Information in DOMINO

Vol.:

No.: 2287

Date: March 24-28,

Pages: 769-771

Verified: OCLC ISSN: 0302-9743 [Format: Serial]

Maxcost: $30.00

Due Date:

Lending Notes:

Bor Notes: