REPRESENTING UNCERTAINTY, PROFILE AND MOVEMENT HISTORY IN MOBILE OBJECTS DATABASES

Eduardo Nóbrega and José Tiago Rolim and Valéria Times
Centro de Informática, Universidade Federal de Pernambuco, Caixa Postal 7851, Recife, PE, Brasil

Abstract: This paper presents a model of data for mobile objects. The model is represented by spatial-temporal classes with mobility aspects, and represents the route and the trajectories where the objects move. The problem of mobile databases inaccuracy is approached. A model of trajectory uncertainty for free route is considered, where spatial operators are defined to capture the uncertainty. Another important concept added by the model is the profile and historical movement, that is, an attempt to increase the precision of prevision for object positions not yet observed. Some simulations results are shown, presenting the efficiency of some techniques for calculating the uncertainty of the positioning prevision.

Key words: Uncertainty; Profile; Movement History; Mobile Objects; Databases.

1. INTRODUCTION

Due to the presence of a new generation of portable machines (*PDAs - Digital Personal Assistants*), to the enormous advance of net technologies and the popularization of the GPS technology (*Global Position System*), researchers and manufacturers started to notice that some applications for mobile devices had started to be viable [4].

There is a wide variety of applications that carry or manipulate objects that change its spatial characteristics through the time. For example: real time applications sensible to localization (i.e. management of car fleets, tracking of trains and aircraft, monitoring of ships) and Applications of

Geographic Information System involving time (i.e. control of the evolution of regions on the time). The domain of these applications currently lacks mobile objects database support [1]. In such applications, data can be distributed in devices that put themselves into motion continuously, generating the need to make this data reflect the current localization of the device. There might be queries within such results depend on the geographic localization, known as LBR - Location Based Retrieval (recovery based on the locality) [2]. In this case, the queries can, besides returning the conventional data on a given object, they must also return the locality of the mobile device. This locality, due the continuous movement, needs to be frequently updated in the database so that it can be related to one instant of time.

A mobile object is any object, punctual or with extension, that changes its geographic position continuously as the time passes and has spatial and temporal characteristics [3]. Currently, it has been a research challenge to solve the problems caused by the continuous movement of mobile objects [5]. Three main reasons for the accomplishment of works associated to the model of data for mobile objects exist. In first place, the traditional databases support the discrete updates of objects in function of spatial and time. That is, the data tend to remain constant until an explicit update modifies its state, but are not well equipped to deal with data that are brought up to date continuously. These continuous updates of the data, if made very frequently, can cause an unsatisfactory performance of the Database Management System. Moreover, one high tax of transmission of objects for the server where the database is located can cause a work overload and high consumption of the width of net band. Thus, due to the continuous movement and net delays the database of mobile objects will sometimes not be able to represent the precise real position. Moreover, it is possible that due to a disconnection, an object cannot bring up to date its position continuously. Thus, alternatives for mobile object modeling have that to be searched so that there is no need to update the database constantly and, besides that, it is still possible to work with not unbalanced data.

In second place, modeling proposals for mobile objects are rare and still very influenced by the traditional modeling of databases [15, 27]. Such models allow to the mobile object storage as structuralized entities, thus allowing to consultations based on its spatial-temporal structure. These objects have a spatial extension that can vary on the time.

In third place, few operators are defined in literature to effectively explore the new possibilities of queries that the mobile objects can offer. What it is necessary in the current applications is a model of data that possess a minimum and robust set of operators, with predicates possessing a high power of expressivity. These types of applications are adjusted for a realistic implementation based on the technology of commercial DBMS.

2. BASICS CONCEPTS

A System of Mobile Database relates to any tool in which the access to the Database is made by mobile stations through a connection wireless [20]. According to Bey Hi *et. al* [3], a spatial-temporal object is an object that possess at least one spatial and one temporal attribute . The other attributes are called conventional or descriptive attributes (that they describe thematic properties of the object). A mobile object can be considered an extension of a spatial-temporal object with mobility characteristics.

The definition of movement profiles is an attempt to increase the precision of the previsions of positions not yet observed of a mobile object. It is assumed that comments can be irregularly spaced in time, of different origins, and that no information of planned route is available. To compute a position and the way expected for a mobile object, a set of observations that inform the position of an object is used. To do so, aspects such as characteristics of objects behavior, until restrictions of the physical environment where such objects can be put into motion are considered.

Descriptions of movement are summaries or successive subgroups of past movements of a particular object and can define a series of typical movements for one determined mobile object. In other words, they are registers of movements frequently made by mobile units [26].

Other important concepts are the deviation and the uncertainty [25]. The deviation is the difference in unit of space between the position of the mobile object and the position prevision by the database system. The uncertainty is the ray around the mobile object in which the object can meet, see figure 1. This circular area represents the space where the object can be contained. Moreover, the uncertainty always comes attached in a reply that involves mobile object positioning. The deviation goes increasing with elapsing of the time until reaching a threshold, that is, the ray of the circumference represented by the uncertainty. When the deviation reaches the threshold is said that the mobile object made a trip and is calculated a new uncertainty based on the data of the previous trip. Another very important concept, on which some models of mobile database are based, is the dynamic attribute [5]. According to Wolfson *et. al* [15], dynamic attributes are attributes that move continuously in function of time, without being explicitly updated.

3. RELATED WORK

Database researches have addressed some aspects related to the modeling and querying of mobile objects localization. One of the important aspects is the access method [6]. There are many recent results trying to solve the problem of indexing the spatial-temporal objects localization and dynamic attributes [7, 8, 9, 10, 11, 12, 13, 14]. An approach about data update politic between mobile units and the server is presented in [28]. In the MOST model, introduced in [15], the notion of dynamic attribute and the representation of mobile objects in function of localization and speed vector are shown. Algebraic specifications of an abstract data type system, constructors and a set of operations are determined in [16, 17, 18].

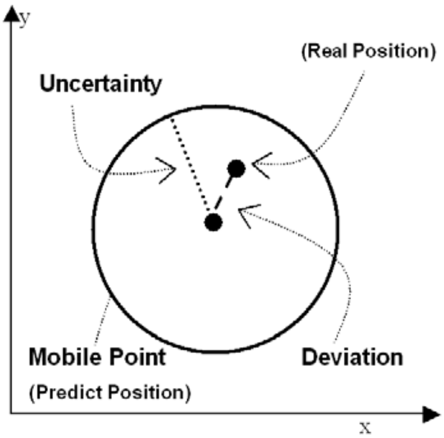

Figure 1. Uncertainty and deviation representation.

About uncertainty, in [19, 20, 25] policies for the update and modeling of inaccuracy are boarded, investigating the trade-off between the costs for communications vs. inaccuracy of the results of queries over mobile objects. Two orthogonal inaccuracy models are described in [21] and [22]. In [21], the bounds of where an object might be located between two points along its route are considered. In the other hand, in [22], a simple model to represent uncertainty is considered, where a circular area around the mobile object represents is possible localization in a given instance of time. The work proposed in [23] presents a simple data model, which integrates the temporal and special uncertainties, using results described in [21, 22]. Moreover, a framework for quantitative values probability in request to the range-queries spatial-temporal is proposed. More recently, the results shown in [24] present a probabilistic model for imprecise data. However, this work considers only one-dimensional ranges (i.e. intervals). In [1], the uncertainty of trajectory is modeled in the form of a cylindrical. Moreover, a set of spatial-temporal operators is introduced, which capture uncertainty and are used to express spatial-temporal range-queries.

4. THE DATA MODEL

The propose data model has as focal point to try to resolve the update problem of mobile object positioning with the database. The aim is to balance the number of updates with the consistency of the database. In other words, the model possess artifices that provide the reduction of the number of updates between the mobile units and the server and exactly thus the database supplies answers with a high degree of consistency. The modeling of geographic data is simple, where a mobile object is represented as a mobile point in the bi-dimensional space related the instant of time. The mobile points must move on road networks, being possible the change of way when passing through a crossing.

The mobile objects has its information stored local in the mobile units, updates of positioning with the server are carried through of continuous way, making the storage of discrete way. The queries can be carried having as target mobile or static objects. Some classes are defined to deal with prevision of movement, improving the answer of the queries carried in instants of times in which the registers had not been updated in the database, predicting where the mobile object would be in that instant of time. The position of the mobile object is stored as a dynamic attribute.

To improve the prediction movement, profiles of movement of objects was created, storing speeds that the object normally keeps in stretches of routes and ways that usually are covered, to facilitate the taken of decision about which way the object would have taken when to exist some possibilities.

The functionalities of profile and movement history are modeled only in the server, this is useful when the connection between client and server is lost or the mobile object does not update the positioning for any another reason. When this situation occurs, it may be necessary to make a query about the mobile object. At the moment that the query is triggered, the positioning information will possibly go as outdated becoming inconsistent the result of the query. Trying to correct this problem, database management system makes a query in the movement history that mobile object to know which route that the object frequently chooses in one stretch, for example, when a mobile object passes through a crossing it can continue in the way or to fold in the crossing starting to cover on another route. Until the moment the position of the object has not updated the result of the queries is based on the movement profile, in addiction, the result of the query will be based on the most likely way that the mobile object chosen.

The data model approach in this work is divided in two modules: a module in the server (figure 3) and another one in the client (figure 4). The module of the client, represented at figure 4, has some more classes and interfaces than the server's module with the purpose of: (i) to deal with the location capture, through some device that supplies information of geographic positioning, and (ii) to calculate the uncertainty, because is depending on the results of these calculations that the client will synchronize its positioning with the server. The server module of the data model provides to the users of the system the ad hoc execution of queries, thus, it is possible the execution of queries based on the localization of mobile objects.

There are some identical classes presented in the two models, this serves to improve the question of the updating politic. Having identical algorithms of movement predicts, the mobile unit goes to know how much the server is predicting missed positioning, updating its position with the server when the prediction reaches a unit deviation, to remember the concept, the deviation is the difference between the real position and the predicted position by the database.

Although the server possesses fewer classes than the client, is in the server that is concentrated the information about all the mobile and fixed objects, demanding a great capacity of throughput to accomplish the queries and storage of information. Due the low capacity of storage of the mobile units, only is stored the attributes of lasts updates. Despite the model of the client seeming denser, the throughput expense is only for calculations of deviation and positioning capitation.

Following, the main classes of the model presented in this work are described. Detaching the classes *EnvironmentMonitor*, *DeviationMonitor*, *AttributeUpdate,* that are responsible for dealing with the uncertainty of the trajectory (see figure 3). And the classes *Profile* and *Path*, that are the classes that represent the profile and the movement history.

Route: Class that represents a route, i.e., the way where the objects move. A route is represented by a list of several segments of line. The end point of a line is the initial point of the following line.

Each route has the following attributes: name, the type of the way (street, avenue or road) and the one-way pointer. The method *getLineSegment* returns a line segment from a given distance covered in the route. The method *pointToDistance* has the function of, given a (x,y) coordinate of an object O and a route R, returns the distance between the object O and the beginning of the route or vice versa. The method *distanceToPoint* is a more simple way to calculate than the point conversion for distance, since, having the distance, we know in which segment of straight line the point is. Thus, it is enough to use similarity of triangles to find the desired point. The method *distanceOfTwoPoints* returns the distance between two points in the route. Finally, the method *routeDistance* returns the route size, covering each one of the segments that constitute the route. The *Route* method is the constructor of the class, receiving the sequence of segments that form the route, besides the properties of the route already mentioned.

As we have stored the points that compose a route, we can thus know if the object is moving in favor or against the route. For this is necessary to know which the current coordinate *(x,y)* of the mobile object and in which position this coordinate if it goes with the sequence of points that compose the route, another necessary information is the final position that the object is found in the last update.

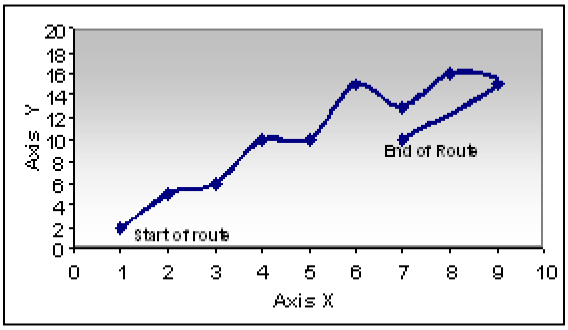

Figure 2. Example of route

Path: This class is modeled in the server's model only. Represents a path commonly used by a mobile point. The path is a set of routes. The method *nextroute* is used to predict the next route when a mobile object passes through a crossing and the server need to predict which the next route to be followed by the mobile object. That is useful for queries in the future time or when the server does not possess the next point that the mobile object is found after pass trough a crossing. An example of query in the future time is "Return the position of object A after 2 minutes from now". The attribute *usualPath* represents the path (set of routes) that is normally used.

LineSegment: Class that represents a line segment. It is composed by a sequence of points. The methods *getStartPoint* and *getEndPoint* supply the initial and final point of the line segment. The method *length* calculates the size of the segment and the method *getPoint* returns a point from the segment given a distance covered in the route.

Point: Class that represents a point in 2D space. It has only two values, *X* and *Y*, that form a pair of coordinates, and are accessible through the methods *getX* and *getY*. The method *point* is the constructor of the class, and it needs the definition of the point coordinates.

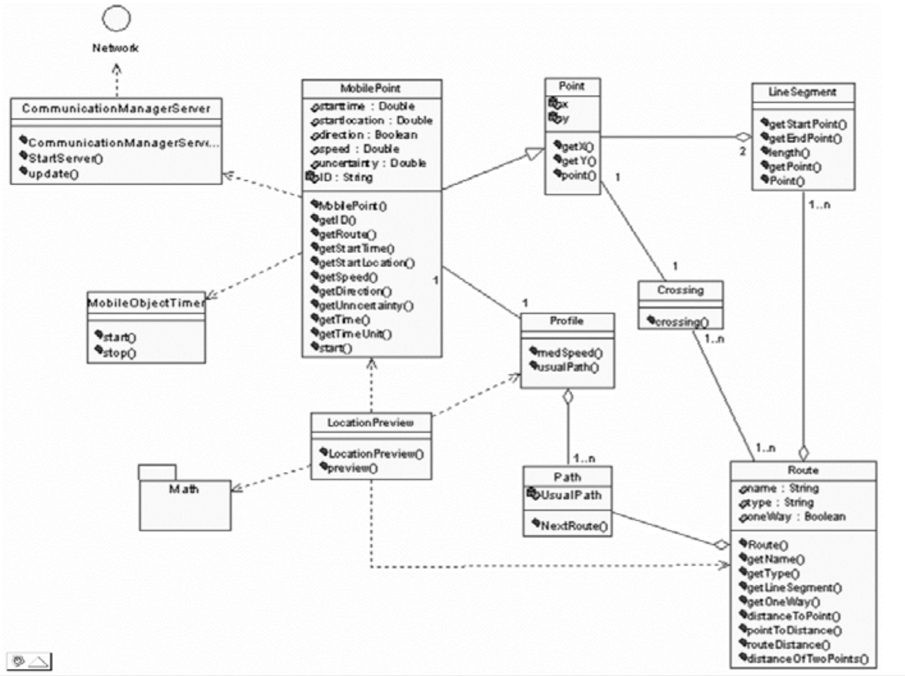

Figure 3. Server's Module.

MobilePoint: Class that represents a mobile object. It inherits the spatial characteristics of the *Point* class, adding mobility properties. Its main methods are: *getDirection* that returns the direction (in accordance or against the route). *getRoute* returns the route in which the object is, during a time *t*. *getSpeed* returns the object speed from a given time *t*. *getStartLocation* returns the initial position in the route. *getStartTime* returns the initial time in the route. *getTime* returns the passed time. *getTimeUnit* returns the time unit. *getUncertainty* gets the uncertainty value when it is updated by the *AttributeUpdate* class. A mobile point has a relationship with the class *profile* (described as follow), with the intention to associate a profile to each mobile point, and, finally, the *Route* class, that indicates which way in the route the object is passing by at the moment.

Profile: All mobile object has a profile, that stores the speeds (attribute *speed*) in which the object uses to keep in definitive stretches and paths (*Path* class) normally covered, in a way to make easy the taking of decision concerning which way the object would have taken when several possibilities exists. It is useful to predict with bigger precision a movement that was still not updated or a position in the future time of a determinate mobile point. It shall be stored standards behaviors, to make the prevision according to reality. These characteristics of movement are taken through the history movement.

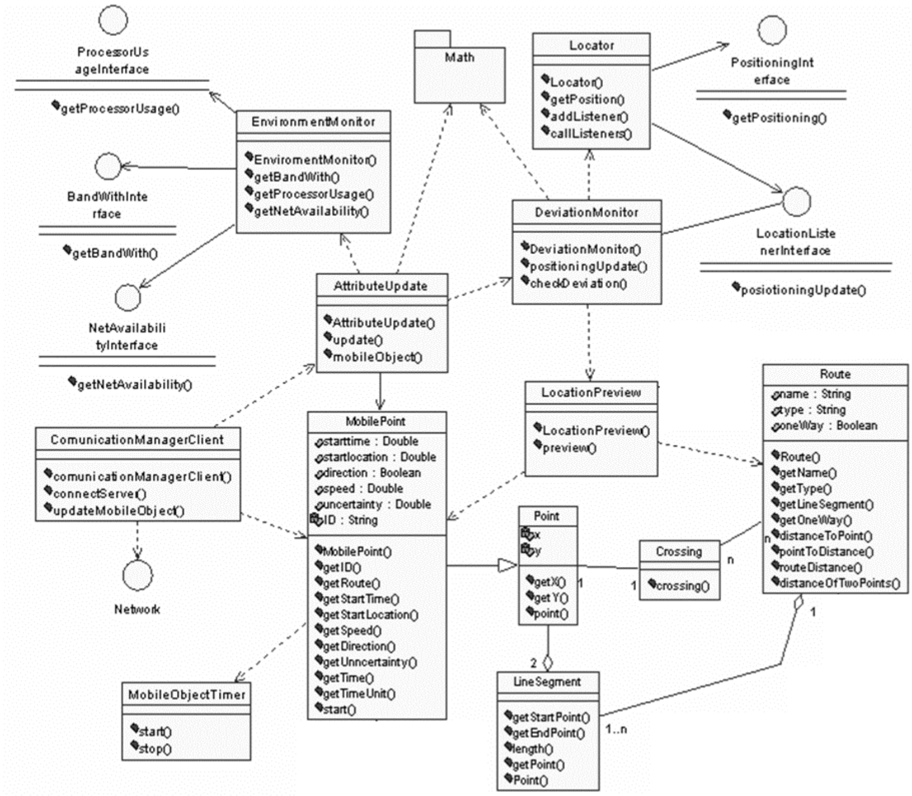

Figure 4. Client's Module.

Cross: It represents a crossing, i.e., it defines a point of intersection between routes and is represented by a point (*x,y*). When passing in a crossing, an object can take the decision to continue in the same route or to pass to another route of the crossing. This class has a relationship with Route class, defining the routes that pass through the crossing, and with the Point class, because a crossing has spatial characteristics of a point. The method *crossing* is the constructor of the class and receives the values from the point coordenates that are the position of the crossing.

DeviationMonitor: The deviation monitor has the function to analyze the information of positioning come from the Locator class, observing every time if the deviation reached the uncertainty. When this happens, it will invoke the *AttributeUpdate* class so that this updates the information of the mobile object. To calculate the deviation, the *DeviationMonitor* has the support of the *LocationPreview*. This class is defined equally in the client and server sides. Therefore, the client is capable to identify how much the server is wrong when answering a positioning query (uncertainty).

LocationPreview: The movement previewer was created to satisfy the queries done in instants of future times or in which the registers have not been updated in the server, so, indicating where the mobile object would be in that instant of time. The movement previewer is composed of a function based on the speed and the time.

***AttributeUpdate*:** The update of attributes has some methods to analyze the behavior of the deviation, network parameters and to realize the processing for computation of the uncertainty. Based in the network conditions and the capacity of processing of the mobile device in the previous travel, the uncertainty is calculated.

***Locator*:** Responsible class for receiving information about the positioning of the mobile devices gifted with a GPS antenna or some external device that repass the datas for the mobile device.

***EnvironmentMonitor*:** The environment monitor receives information as: level from processing, band width and availability of network and supplies its data to the attribute update.

CommunicationManagerClient: The communication manager is responsible for making the communication between the client and the server, and after that, sends the attributes that had been modified by the attribute update to the server. It must check if already a communication exists. In case that already exists, keeps it. In case that contrary, it opens a new.

Math Package: The mathematical package includes some classes, responsible by accomplishment of mathematical calculations. To simplify the model the classes will not be detailed. This package aids the attribute update and the deviation monitor. The classes that belong to this package are: Function approximator, Integrals Calculator and Converter of points and distances.

5. OBTAINED RESULTS

To make the simulation of the data model, some interfaces of environment management had been implemented (Localizer, Network Availability and Processor Usage). The implementation of the interfaces result in a Environment and Movement Simulator, see figure 5.

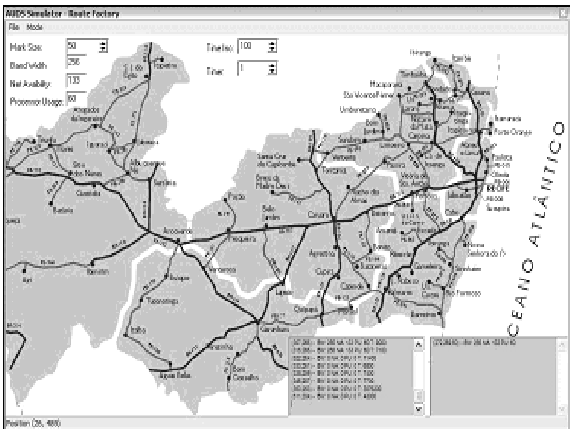

Figure 5. Movement Simulator.

The mobile objects move through roads in the map of the State of Pernambuco. The simulator reads a data file containing information about positioning, band width, network availability, processor usage and an interval of time for the next reading. The format of the file is as follows:

```
<Position X>, <Position Y>, <Band width>, <Availability>, <Processor
usage>, <Time for the next reading>
```

Observes the cutting of an example of an input file for simulation of movement:

```
1: 275,270,256,133,60,1100
2: 276,270,256,133,60,1700
3: 277,270,256,133,60,3200
4: 278,270,256,133,60,1500
5: 279,270,256,133,65,2900
6: 280,272,256,133,62,1500
7: 281,273,256,133,61,1800
8: 282,275,256,133,69,1700
9: 283,277,256,133,50,1600
```

In line 1, the mobile object is found in the position (275,270), the band width is 256Kbps, the availability of such band is of 133Kbps, the processor usage is in 60% and the simulator will have to wait 1100ms to read the next line.

The figure 6 shows an example of a route in the map. In this example, the mobile object moves from Arco Verde city until Pesqueira city. This route was predefined by the driver of a mobile unit (i.e. an automobile). Before initiating a travel the stretch was determined to be covered by the mobile object. The experiments had been performed in a prototype of a system that is based on the model considered in this work.

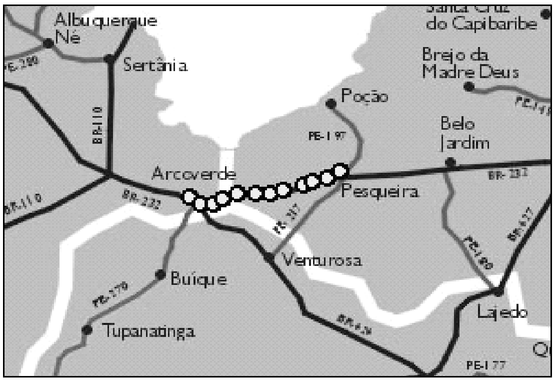

Figure 6. Arco Verde-Pesqueira Route.

The speed of the mobile object if kept varied in relation to the time during the passage of the route, as it can be seen in figure 7.

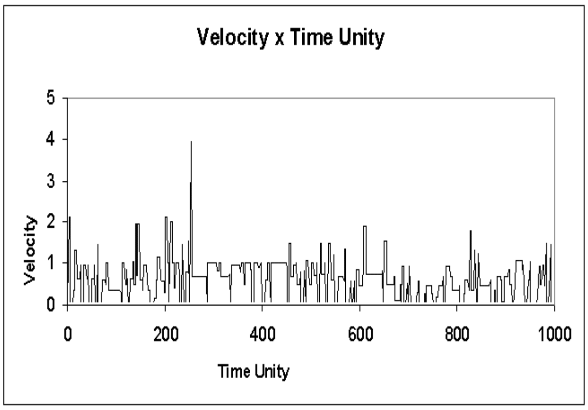

Figure 7. Speed Variation by Unit of time.

The Figures 8 and 9 show graphs, respectively, that they represent the variation of the deviation and the uncertainty for unit of time. Comparing figure 7 with figure 8, we can notice that the bigger the speed, the faster the deviation will reach the uncertainty.

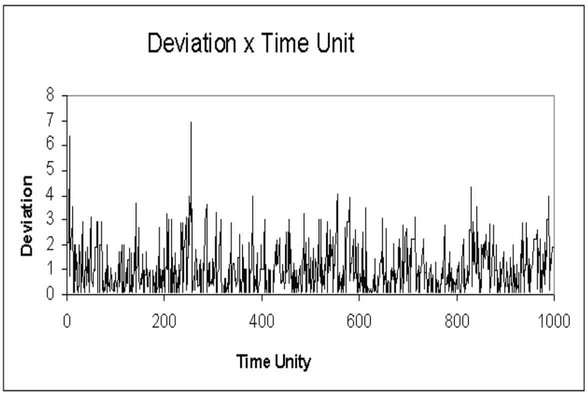

Figure 8. Deviation variation by Unit of time.

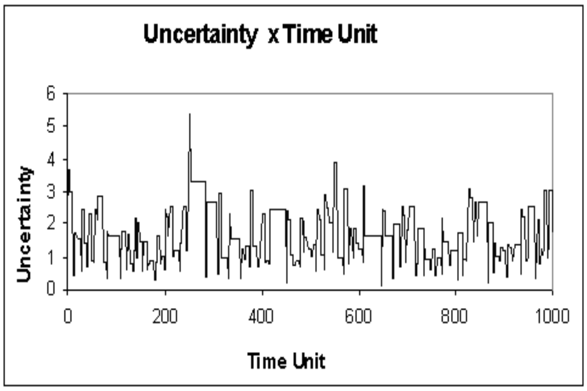

Figure 9. Uncertainty Variation by Unit of time.

The Figure 10 shows the efficiency of the positioning prevision of mobile objects. The lines of real and predicting position practically are overlapped, showing that the used technique for positioning prevision is efficient. In other words, the difference between the real position and predicted position is small.

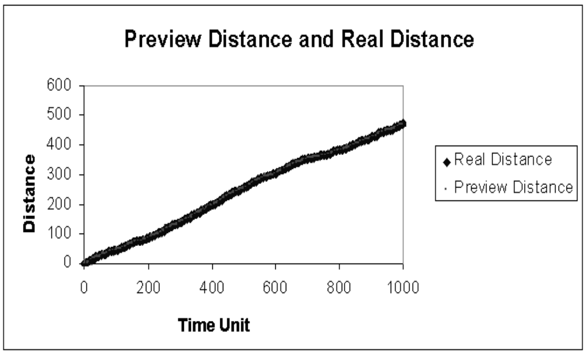

Figure 10. Comparison between the previewed position and the real position.

6. CONCLUSIONS AND FUTURE WORK

In this work some concepts and problems related the modeling for mobile objects database and the tendency for its solutions had been described.

A conceptual data model for mobile objects was proposed, focusing: (i) the position prevision based on the history movement and (ii) the trajectory uncertainty modeling. The approach objective is to improve the movement prevision result in applications where the continuous movement of mobile objects is important.

The use of movement prevision introduced in [20], adding the techniques of update introduced in movement behavior standards study proposal in [26], can bring a performance gain to the applications that needs the mobile objects to be constantly updating its position to guarantee the query result precision.

Currently, some operators are being defined as complement of this work, to express queries that involve mobile objects. The profile implementation and movement history to improve the prevision movement when a mobile object goes through a crossing, and does not update your position, is in progress.

The researches about mobile objects and the functionalities that can be implemented support a series of future works. From the results taken in the model initial implementation considered in this work, some important questions for its improvement and extension can be identified. It is important to deal with other geometrics for mobile objects in addition to point. Starting from the definition of a basic set of operators, a language that explores the mobile objects spatial-temporal characteristics can be defined.

ACKNOWLEDGMENTS

We are grateful to our project partners for asking many of the questions addressed in this paper. Thanks for many clarifying discussions especially to Thiago Costa, Rômulo Augusto, Juliana Xavier, Otávio França and Edson Lima.

REFERENCES

[1] G. Trajcevski, O. Wolfson, K. Hinrichs and S. Chamberlain, "Managing Uncertainty in Moving Objects Databases", *ACM Transactions on Database Systems (TODS)*, (2004).

[2] H. Dunham and V. Kumar, "Location Dependent Data and its Management in Mobile Databases", *Proceedings of the Ninth International Workshop on Database and Expert Systems Applications* (1998), 414 – 419.

[3] B. Yi and C. Medeiros, "Um modelo de Dados para Objetos Móveis". *IV Simpósio Brasileiro de GeoInformátic* (2002), 33–40.

[4] GSM World - The *world wide web site of the GSM Association,* http://www.gsmworld.com/technology/applications/index.shtml

[5] R. S. Oliveira, "Modelo e implementação de Objetos Móveis em Banco de Dados Objeto-Relacional", *Master´s thesis, Universidade Federal de Campina Grande,Paraíba, Brazil*, (2003).

[6] V. Graede and O. Günther, "Multidimensional access methods". *ACM Computing Surveys 11(4)*, (1998).

[7] A. K. Agarwal, L. Arge and J. Erickson. "Indexing moving points", *ACM PODS Conference 19*, 2000.

[8] D. Kollios, D. Gunopulos and V. J. Tsotras, "On indexing mobile objects". *ACM PODS Conference 18*, (1999).

[9] D. Gunopulos, G. Kollios and V. J. Tsotras, "Nearest neighbour queries in a mobile environment", *STDBM*, (1999).

[10] C. Jensen, D. Pfoser and Y. Theodoridis, "Indexing trajectories of moving point objects", *Technical Report, Dept. of Computer Science, University of Aalborg*, (1999).

[11]C. S. Jensen, M. A. Lopez, S. T. Leutenegger and S. Saltenis, "Indexing the positions of continuously moving objects", *Technical Report TR - 44, TimeCenter*, (1999).

[12] J. Tayeb, O. Ulusoy and O. Wolfson, "A quadtree – based dynamic attribute indexing method", *The Computer Journal 41(3)*, (1998).

[13] A. N. Papadopoulos, Y. Manolopoulos, Y. Theodoridis and T. Sellis, "Specifications for efficient indexing in spatiotemporal databases". *IEEE SSDBM*, (1999).

[14] M. A. Nascimento, Y. Theodoridis and J. R. O. Silva , "On the generation of spatiotemporal datasets". *Int'l symposium on Large Spatial Databases 6*, (1999).

[15] A.P.Sistla, O.Wolfson , S.Chamberlain and S.Dao, "Modeling and querying moving objects". *Proceedings IEEE Intl. Conf. On Data Engeneering, Birminghan, UK* (1997), 422–432

[16] M. Erwig, M. Schneider and R. H. Güting, "Temporal and spatio – temporal datasets and their expressive power". Technical *Report, Informatik berichte* (1997), 225-12.

[17] E. Nardelli, M. Schneider, L. Forlizzi and R. H. Güting, "A data model and data structures for moving objects databases". *ACM SIGMOD*, (2000).

[18] C. Jensen, M. H. Böhlen, M. Erwig, M. Schneider, M. Vazirgiannis, N. Lorentzos and R. H. Güting, "A foundation for representing and queirying moving objects". *ACM Transactions on Database Systems (TODS)*, (2000).

[19] G. Mendez, L. Jiang, O. Wolfson, S. Chamberlain and S. Dao, "Cost and imprecisionin modeling the position of moving objects". *ICDE 14*, (1998).

[20] A. P. Sistla, Y. Yesha, O. Wolfson and S. Chamberlain, "Updating and Querying Databases that Track Mobile Units", *Special issue of the Distributed and Parallel Databases Journal on Mobile Data Management and Applications", Kluwer Academic Publishers 7(3)* (1999), 257-287.

[21] C. Jensen and D. Pfoser, "Capturing the uncertainty of moving objects representation". *SSDB*, (1999).

[22] F. Zhang, G. Trajcevski, O. Wolfson, and S. Chamberlain, "The geometry of uncertainty in moving objects databases". *EDBT*, (2002).

[23] G. Trajcevski, "Probabilistic Range Queries in Moving Objects Databases with Uncertainty". *MobiDE'03, San Diego, California, USA*, (2003), 39-45.

[24] D. Kalashnikov, R. Cheng, and S. Prabhakar, "Evaluating probabilistic queries over imprecise data". *ACM SIGMOD*, (2003).

[25] G. Mendez, L. Jiang, O. Wolfson, S. Chamberlain and S. Dao, "Cost and imprecisionin modeling the position of moving objects". *ICDE 14*, 1998.

[26] B. Kate and P. Mustafa. "Estimating positions and paths of moving objects". Workshop *Proceedings IEEE Press* (2000), 155-162.

[27] M. Vazirgiannis and O. Wolfson. "A spatiotemporal model and language for moving objects on road networks". *SSTD*, (2001).

[28] T. Brinkhoff and J. Weitkämper. "Continuous Queries within an Architecture for Querying XML-Represented Moving Objects". SSTD, (2001).