Extending UML for trajectory data warehouses conceptual modelling

Wided Oueslati[#1], Jalel Akaichi[*2]

Computer Science Department, Higher Institute of Management
Tunisia

***ABSTRACT*— The new positioning and information capture technologies are able to treat data related to moving objects taking place in targeted phenomena. This gave birth to a new data source type called trajectory data (TD) which handle information related to moving objects. Trajectory Data must be integrated in a new data warehouse type called trajectory data warehouse (TDW) that is essential to model and to implement in order to analyze and understand the nature and the behavior of movements of objects in various contexts. However, classical conceptual modeling does not incorporate the specificity of trajectory data due to the complexity of their components that are spatial, temporal and thematic (semantic). For this reason, we focus in this paper on presenting the conceptual modeling of the trajectory data warehouse by defining a new profile using the StarUML extensibility mechanism.**

***Keywords—Trajectory data; trajectory data warehouse; UML extension; trajectory UML profile*.**

I. Introduction

The success of the warehousing process rests on a good conceptual modeling schema. In fact, conceptual modeling offers a higher level of abstraction while describing the data warehousing project since it stays valid in case of technological evolution. Besides, it allows determining analysis possibilities for the warehouse. However, no contribution is at the present time standard in term of trajectory data semantic models. This finding leads us to propose a new UML profile with user oriented graphical support to represent trajectory data and trajectory data warehouse conceptual modeling with structural model (class diagram) and dynamic model (sequence diagram).

This paper is organized as follows. In section 2, we present an overview of research works related to conceptual approaches and extensibility of UML for applications' needs. In section 3, we present the methodology that we adopted to extend the StarUML profile. In section 4, we present the Trajectory UML profile. In section 5, we present the trajectory UML profile realization. In section 6, we summarize the work and we propose some perspectives that can be done in the future.

II. Related Works

In this section, we present different approaches related to the conceptual modeling methodology, then we present research works that extended UML to adopt it to their conceptual modeling needs. In the literature, we can find three categories of conceptual approaches; the top down approach, the bottom up approach and the middle out approach. The difference between those latter is situated in the starting point. In fact, each approach has its own starting point such as users' needs, data marts or both users' needs and data marts. Concerning the top down approach, this latter has to answer users' requirements without any exception. It is very expensive in term of time since it requires the whole conceptual modeling of the DW as well as its realization and it is difficult because it requires the knowledge in advance of dimensions and facts [1]. In this category, authors of [2] present a Multidimensional Aggregation Cube (MAC) method. This latter insures the construction of a multidimensional schema from the definition of decision makers' needs but the defined schema is partial because it describes only the hierarchies of dimensions. The goal of MAC is to supply an intuitive methodology of data modeling used in the multidimensional analysis. It models real world scenarios using concepts which are very similar to OLAP.

In MAC, data are described as dimensional levels, drilling relationships, dimensions, cubes and attributes. Dimension levels are a set of dimension members. Those latter are the most detailed modeling concepts and they present real world instances' properties. Drilling relationships are used to present how one level element can be decomposed of other levels' elements. The dimension paths present a set of drilling relationships which are used to model a significant sequence of drill down operations. Dimensions are used to define a significant group of dimension paths. This grouping is essential to model semantic relationships. Cubes are the only concept which associates properties' values with real measures' values. They insist on the complex hierarchy structure defined by dimensions. The top down approach can be used in the Goal-driven methodology [3]. In fact, this latter focuses on the company's strategy by occurring the executives of the company. For the bottom up approach, this latter consists on creating the schema step by step (data marts) until the obtaining of a real DW [1].

It is simple to be realized but it requires an important work in the data integration phase. Besides, there is always the risk of redundancy due to the fact that each table is created independently. Authors in [4] present a dimensional fact model. This latter relays on the construction of data marts firstly. This can insure the success in case of complex projects but it neglects the role of decision makers. Authors in [5] adopt the bottom up approach. In fact, they present a dimensional model development method from traditional Entity-Relationship models to insure the modelling of DWs and Data Marts. This method is based on three steps: the first step includes the classification of data models' entities into a set of categories. This leads to the production of a dimensional model from an Entity-Relationship model. We find the transactional entities that insure the storage of details concerning particular events in the company. We find also the component entities that are directly connected to transactional entities through the 1..* relationship. Those entities allow defining details of each transaction. Classification entities are connected to component entities through the 1..* relationship. Classification entities present the existing hierarchies in the model.

The second step consists on identifying hierarchies that exist in the model. In fact, the hierarchy is an important concept in the dimensional modelling level. The third step consists on grouping hierarchies and aggregations together to form a dimensional model. At this level, we find two operators that are used to product models of dimensions. In fact, the first operator can transform the high level entities to low level entities. This can be done until the arrival at the bottom of the architecture. The aim is to have an only table at the end. For the second operator, it is applied on transactional data to create a new entity which contains summarized data. This approach is used as a base for the data driven and user driven methodologies. In fact, as presented in [3], the data driven (supply driven) methodology starts by analyzing operational data sources to identify existent data. Users' intervention is limited to the choice of necessary data for the decision making process. This methodology is adopted when data sources are valid.

For the user driven methodology, it starts by collecting users' needs. Those needs will be integrated in order to obtain one multidimensional schema. This approach is appreciated by users but it presents a big challenge. In fact, managers of projects must be able to take into account the different points of views. For the middle out approach, it is an hybrid method since it benefits from the two approaches cited above. Authors in [6] present an example of hybrid modelling method that is based on the top down and the bottom up approaches. The bottom up approach is based on three steps: the collection of needs, the specification and the formalization of those needs in the form of multidimensional constellation schema. The top down approach includes the data collection and the construction of a multidimensional schema that allows decision making. The approach is based on the description of decision makers needs. Those two approaches allow having two schemas, then from those latter only one schema will be derived and kept. The middle out approach is composed of four phases; the users' needs analysis, the confrontation/comparison, the resolution of conflicts and the implementation.

Authors in [7] present another method which uses the middle out approach. This latter is based on three steps: the collection of users' requirements by the top down approach, the recovery of star schema by the bottom up approach and finally the integration phase. This latter connects the obtained star schema from the first step to the obtained star schema from the second step. The integration is realized thanks to a set of matrix. Users' requirements are collected by the Goal Question Metric (GQM) paradigm. This latter allow attributing metrics to identified goals. This facilitates the filtering and the deletion of not useful goals. Authors of [7] consider that the modeling of warehouses is a process based on goals, and then users' goals related to DW development will be present explicitly. Goals will be analyzed in order to reduce their number (authors take into account the similarity of goals). For the choice of star schema, authors use the Entity-Relationship model. This latter is exhaustively analyzed to find entities that will be transformed to facts and dimensions. The transformation process of Entity-Relationship model to a star schema is based on three steps. The first step is the construction of a connected graph that serves to synthesized data. The second step is to extract a snowflake schema from the graph. The third step is the integration phase. In fact, authors exploit the structure of the warehouse of the first phase and the set of possible schemas of the second phase, and then they apply a set of steps such as converting of schema to express them with the same terminology. Within UML-based conceptual models, the most famous approaches are of Trujillo and his team.

In [8], authors proposed UML extensions for object-oriented multidimensional modeling. This extension is performed thanks to stereotype mechanism, tagged values and constraints expressed in OCL-Object Constraint Language, in addition to a set of Well-Formedness rules managing new elements added and determining the semantic of the model. Stereotypes and icons allow an expressive representation of different constituent elements of a multidimensional model namely fact classes, dimension classes, hierarchy levels and attributes. Dimension level classes (stereotyped base classes) should define a directed acyclic graph rooted in dimension class. Concerning relationships, the aggregation links facts to dimensions, and association/generalization links dimension levels (having Base stereotype) between each other's.

In another work of the same team [9], an UML package is proposed to facilitate modeling of large data warehouse systems. In fact, they suggest a set of UML diagrams (package) extended with the aforementioned stereotypes, icons and constraints (OCL) to cope with multidimensional modeling and consequently designers will not be limited only to the class diagram.

Several works [10] [11] [12] [13] [14] [15] proceeded by UML profiles to represent their models. In fact, in [11] the proposed profile is used for the oriented agent modeling. In [12] the authors represented a profile for the mobile systems conception. In [13], authors propose a profile for the modeling of association rules of data mining. In [14] authors propose a profile to model data mining with the temporal series in the data warehouse. In [15], authors extended UML to introduce new stereotypes and icons to handle spatial and temporal properties at the conceptual level. This led to the visual modeling tool so called Perceptory.

III. Adopted Methodology

There are three methods of conceptual modeling of DWs; the first one is the top down approach [16] that is based on the needs of the users, the second is the bottom-up approach [5] that begins with the operational data sources and finally the mixed approach [17] that combines the two previous approaches. We used the top down approach in our modeling phase because we were interested in user's needs. In term of MDA (Model Driven Architecture) [18] our solution is situated in the CIM (Computation Independent Model) level because the models are not inevitably transformed into code. For the abstraction levels [19] (conceptual, logical and physical) our solution is established to cover the conceptual level. Here is a plan showing the position of our solution:

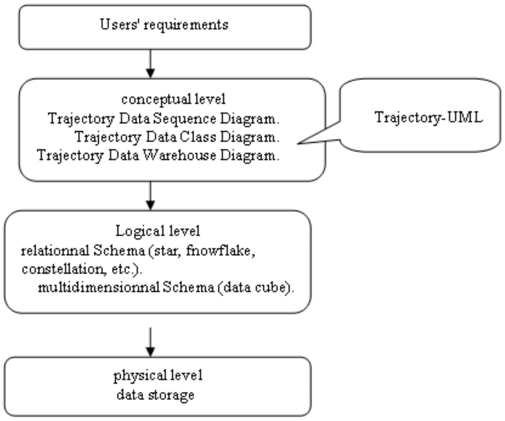

Users' requirements
conceptual level
Trajectory Data Sequence Diagram.
Trajectory Data Class Diagram.
Trajectory Data Warehouse Diagram.
Trajectory-UML
Logical level
relationnal Schema (star, fnowflake, constellation, etc.).
multidimensionnal Schema (data cube).
physical level
data storage

Fig 1. Our solution's position

We also adopted object oriented paradigm because it has several advantages for the multidimensional modeling such as the classification / instantiation, the Generalization / specialization and the Aggregation / decomposition. We chose to adopt the object oriented approach which is based on the UML profiles. We are inspired by the work of [10] to follow the mechanism of profile UML to model the multidimensional part and the work of [20] to widen this profile to the trajectory data and spatiotemporal data.

IV. TRAJECTORY UML PROFILE

With the expansion of technology based on captors (GPS, RFID, etc), it became possible to establish new information systems involving moving objects, to set their trajectories and to pusue their movements. The Trajectory Data Warehouses (TDWs) emerge for needs to study the devices of moving objects in order to develop the decision process. It is necessary to provide a formal representation to TDWs, to understand the concrete world and for a good human comprehension to moving objects phenomena. Modelling TDWs is in its early stage. The classic approaches do not have their good productions for a clear and standard methodology given the great complexity of trajectory data and the difficulty to model varying fields.

An UML profile [21] allows specializing UML in a precise domain, it consists of stereotypes, tagged values and constraints. A stereotype [22] is an element of the model that defines new values, new constraints and a new graphic representation. Its role is to give a semantic representation to an element of the model. A stereotype can be represented as a string character between two quotation marks << >> or with an icon. A marked value specifies a new property attached to an element of the model. It is represented between {} and placed with the name of another element. A constraint can become attached to any element of the model to refine its semantics and prevent an arbitrary use of the various elements. It can be defined with the natural language and\or with the OCL (object constraint language) [21] which is a declarative language that allows developers to write constraints on the model's objects. Recently, UML profiles have a great progress in the ways for conception of Data Warehouses. We present in this section, a conceptual solution for trajectory data warehouses design. We proceeded by an UML profile in order to add stereotypes, tagged values and constraints. Our Trajectory UML profile contains two diagrams: the first one is Trajectory Data Class Diagram. This latter has for purpose to model the trajectory data of the moving objects. The second diagram is Trajectory Data Warehouse Diagram. This latter represents the TDWs in a multidimensional context.

A. Trajectory Class Diagram

We defined in this diagram stereotypes and icons related to trajectories such as moving object, stop, move, trajectory section, pda, gps and location. This diagram can be used in each case based on trajectories of moving objects.

1) Classes' Stereotypes: We defined in this table classes's stereotypes used in the modeling

TABLE I : STEREOTYPES DESCRIPTION AND REPRESENTATION

Stereotype name	Class type	Description	Icon
<< movingobject>>	Class	This stereotype indicates that the class represents a moving object. In our case study the moving object is the mobile hospital	<<movingobject>> mobile hospital2
<< trajectory>>	Class	This stereotype indicates that the class represents a trajectory	<<trajectory>> trajectory2
<<trajectorysection>>	Class	This stereotype indicates that the class represents a part of trajectory "trajectory section" that is a component of the trajectory	<<trajectorysection>> tr-section2
<< stop>>	Class	This stereotype indicates that the class represents the "stop" which is a component of the trajectory	<<stop>> stop2 ●
<< move>>	Class	This stereotype indicates that the class represents the	<<move>> move2

		movement "move" which is a component of the trajectory	

At the end of this section, we propose the following example of defining an UML Class related to the Trajectory Data Class Diagram with an extended stereotype and icon using XML:

```
<STEREOTYPE>
<NAME>movingobject</NAME>
<DESCRIPTION></DESCRIPTION>
<BASECLASSES>
<BASECLASS>UMLClass</BASECLASS>
</BASECLASSES>
 <ICON minWidth="30" minHeight="20">mh.bmp</ICON>
<SMALLICON>mh.bmp</SMALLICON>
<RELATEDTAGDEFINITIONSET>movingobject</RELATEDTAGDEFINI
TIONSET>
```

2) Association' Stereotypes: For associations we kept the standard elements of UML such as the association, the generalization, the aggregation and the composition because we noticed that these relations meet users' needs. Besides, we added specific associations that can exist between different components of trajectories as described in the following table:

TABLE III
ASSOCIATIONS STEREOTYPES DESCRIPTION AND REPRESENTATION

Stereotype name	Class type	Description	Icon
<< Inside>>	Association	In our case study the class stop is Inside a given spatial dimension called location	◉
<< Adjacency>>	Association	Two adjacent trajectory sections share the same stop.	○●
<<Cross >>	Association	A moving object cross a trajectory or a trajectory section.	⌒●

At the end of this section, we propose the following example of defining an UML Association related to the Trajectory Data Class Diagram with an extended stereotype and icon using XML:

```
<STEREOTYPE>
<NAME>Inside</NAME>
<DESCRIPTION></DESCRIPTION>
<BASECLASSES>
<BASECLASS>UMLAssociation</BASECLASS>
</BASECLASSES>
 <ICON minWidth="30" minHeight="20">Inside.bmp</ICON>
<SMALLICON>Inside.bmp</SMALLICON>
<RELATEDTAGDEFINITIONSET>Inside</RELATEDTAGDEFINITIONS
ET>
</STEREOTYPE>
```

V. TRAJECTORY UML PROFILE REALIZATION

To implement our approach we chose the StarUML open source platform that uses the language XML to create the profiles UML. In this section we describe StarUML by showing its stretchable parts, and then we model a trajectory and their components with our Trajectory-UML profile.

B. The StarUML platform

StarUML is a modeling platform with the UML language, conceived to support the MDA (Model Driven Architecture) approach. It is characterized by a strong flexibility and an excellent extensibility of its features. Indeed, besides the predefined functions, StarUML allows the addition of new functions which can be adapted to the user's needs. The inconveniences of this platform are that it does not allow specifying more than a stereotype for an element and it excludes the definition of the constraints. Thus in our work we considered that every element has only a single stereotype.

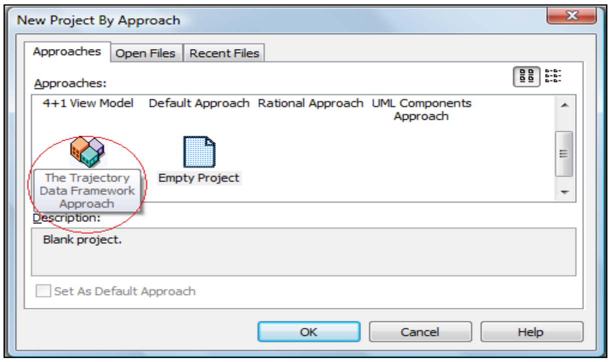
C. The implementation of Trajectory-UML profile

An UML profile is one package belonging to the mechanism of extension. This package is stereotypical < < Profile > > which is written in XML as we see in the following figure:

```
<? xml version="1.0" encoding="UTF-8" ?>
<PROFILE version="1.0">
<HEADER>
<NAME>TDW</NAME>
<DISPLAYNAME>Data Modeling</DISPLAYNAME>
<DESCRIPTION>TDW Data Modeling Profile</DESCRIPTION>
<AUTOINCLUDE>True</AUTOINCLUDE>
</HEADER>
```

In the StarUML platform, we added an approach named "The tdw Framework" and a profile UML called "TDW model" that contains two diagrams which are respectively; "Trajectory Data Class Diagram" and "Trajectory Data Warehouse Diagram". Indeed, we have created two files XML one for the approach and the other one for the profile. Inside these files we appealed to extensions of notation which are files written in Scheme language (Dialect of LISP) which allows realizing specific notations that are different from those contained in UML.

In this part, we represent the interfaces of our added approach to the platform and the various realized diagrams. When we start StarUML, the dialog box "New Project by approach" appears to choose the wished approach. Below we find our approach called "tdw Framework ".



Blank project.

Fig 2. Trajectory data warehouse UML approach

To add diagrams to these models we click the straight button of the mouse, we choose "Add Diagram" and we find the diagrams of our approach. The following figure shows how this step takes place:

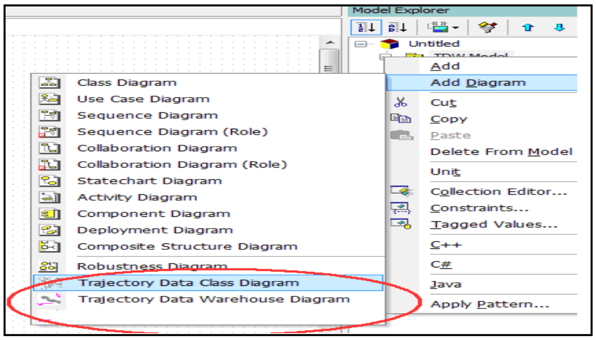


Fig 3. Diagrams of TDW-UML

For every type of diagram we added a palette. These latter allows to visualize the stereotypes and icons of each diagram and to use them. In our case, we created three palettes. The first one is related to the Trajectory Data Warehouse Diagram, the second one is related to the Trajectory Data Class Diagram.

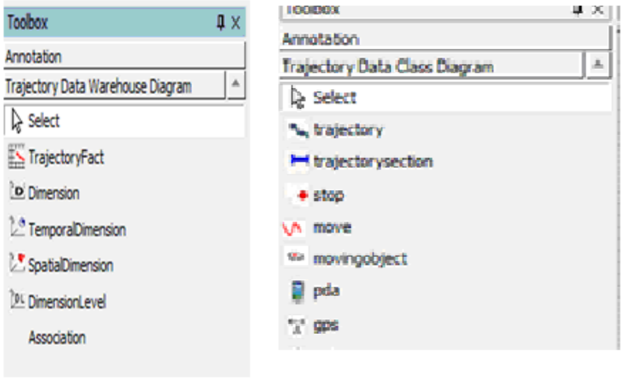


Fig 4. The added pallets

D. Validation of trajectory-UML functioning

Trajectory-UML is guided by some objectives such as the possibility to describe explicitly relationships between trajectories and its components (trajectory-section, stops and moves). Those relationships can be of different types such as topologic, metric, aggregation. Besides, Trajectory-UML allows modeling classical data by offering a well known set of concepts such as class, attribute, association, generalization, composition. From the ergonomic point of view, the trajectory and its components are visualized in diagrams by pictograms and stereotypes. This allows an immediate unambiguous apprehension of additional features. In this section we model the concept of trajectories and their components with our Trajectory-UML profile.

We created a new diagram called Trajectory Data Class Diagram, in which we added some pictograms and some stereotypes to identify each class (entity). To do this, we used some pictograms of MADS project [23]. The same idea was done in [20] but in a relational model.

In this diagram, there are some stereotypes and icons that can be used in any application related to moving objects. In fact, each moving object has a trajectory. This latter is composed of trajectory sections that are composed of moves and stops. Those latter are in a given location. For a generic Trajectory Data Class Diagram, we propose to keep the classes: Moving object, Trajectory, Trajectory-section, Move, Stop and Location. In the following, we propose the Trajectory Data Class Diagram with Trajectory-UML.

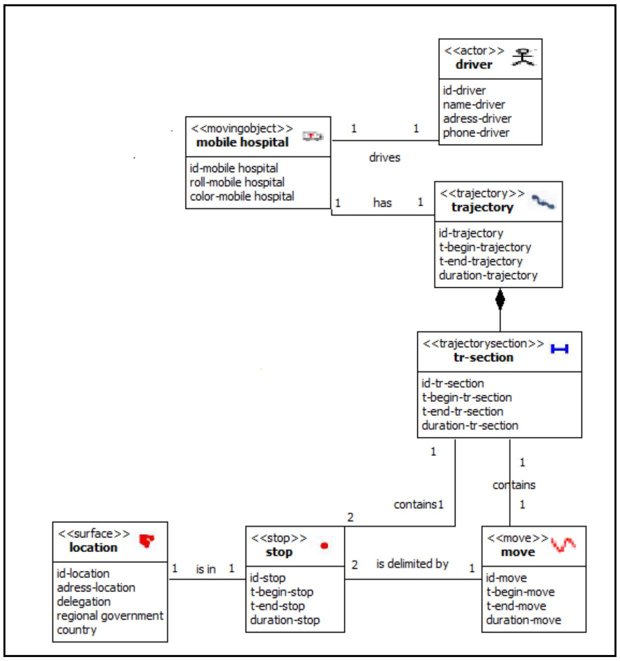


Fig 5. The trajectory data class diagram with our TDW profile

VI. Conclusion

In this paper, we described our profile named Trajectory-UML. This profile contains the diagram "Trajectory Data Class Diagram" which gives a conceptual representation of the trajectory of a moving object by specifying relationships between different entities of the diagram. We described the realization of the Trajectory-UML profile. To estimate our approach we ended this paper with an experimentation of the trajectory data class diagram. We propose as future work to represent a model of the component diagram that is based on an UML profile for the physical level.

REFERENCES

[1] E. wayane,. Four ways to build a data warehouse. 2003.

[2] T. Karayannidis, N. Sellis, T. MAC: Conceptual data modeling for OLAP, Proc of the International Workshop on DMDW. 2001.

[3] M, Golfarelli,. From user requirements to conceptual design in data warehouse design. Information science reference, 2009.

[4] M. Rizzi, S,. WAND: A case tool for workload based design of a data mart. In Proc SEBD, Portoferraio, Italy, pp. 422-426, 2002.

[5] D. Moody, M. Kortink,. From enterprise models to dimensional models: A methodology for data warehouse and data mart design. In Proc.2nd DMDW,Stockholm, Sweden, 2000.

[6] J. Ghozzi, O. Teste,. Méthode de conception d'une base multidimensionnelle contrainte, 1iere journée francophone sur les entropots de données et l'analyse en ligne, toulouse, pp. 51-70, 2005.

[7] Bonifati, A. Cattaneo, F. Designing data marts for data warehouses. ACM transactions on software engineering and methodology, pp. 452-483, 2001.

[8] Lujan-Mora, S., & Trujillo, J. (2002). Extending UML for Multidimensional Modeling. Alicante University, Spain.

[9] Lujan-Mora, S., Trujillo, J., Song, Y. (2002). Multidimensional Modeling with UML Package Diagrams.

[10] S. Lujàn-Mora, J. Trujillo and I. Yeol Song. "A UML profile for multidimensional modelling in data warehouses". Data & Knowledge Engineering (DKE), pp. 725-769. 2006.

[11] G. Wagner, Eindhoven. "A UML Profile for Agent-Oriented Modeling". Eindhoven Faculty of Technology Management, http://tmitwww.tm.tue.nl/staff/gwagner. 2002.

[12] V. Grassi, R. Mirandola and A. Sabetta. "A UML Profile to Model Mobile Systems". Universit`a di Roma "Tor Vergata", Italy. 2004.

[13] J. Zubcoff and J. Trujillo. "A UML 2.0 profile to design Association Rule mining models in the multidimensional conceptual modeling of data warehouses". Data & Knowledge Engineering, pp. 44–62. 2007.

[14] J. Zubcoff, J. Pardillo and J. Trujillo "A UML profile for the conceptual modelling of data-mining with time-series in data warehouses", Information and Software Technology, pp. 977–992 Contents lists, 2009.

[15] J .Brodeur, Y .Bédard, M.J. Proulx. Modelling Geospatial Application Databases using UML-based Repositories Aligned with International Standards in Geomatics. In Proc. of the ACM Symposium on Advances in Geographic Information Systems, ACM GIS, 2000.

[16] N. Tryfona, F. Busborg et J.G. Christiansen. "starER: A Conceptual Model for Data Warehouse Design". Dans: Proceedings of the ACM 2nd International Workshop on Data warehousing and OLAP (DOLAP'99), Kansas City, Missouri, USA, pp. 3–8. 1999.

[17] C. Sapia, M. Blaschka, G. Höfling, B. Dinter. "Extending the E/R Model for the multidimensional paradigm". International Workshop on Data Warehousing and Data Mining (DWDM '98) in conjunction with the 17th Int. Conf. on Conceptual Modeling (ER '98), (Singapore). 1998.

[18] J. Norberto, M. Juan Trujillo. "An MDA approach for the development of data warehouses ". Decision Support Systems, pp. 41–58. 2008.

[19] N. Prat, J. Akoka and I. Comyn-Wattiaun. "A UML-based data warehouse design method". Decision Support Systems, pp. 1449–1473. 2006.

[20] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. de Macedo, F. Porto, C. Vangenot. A Conceptual View on Trajectories. Research Report. Ecole Polytechnique Fédérale, Database Laboratory, Lausane, Switzerland. May, 29th, 2007.

[21] Object Management Group (OMG). "Unified Modeling Language Specification 1.5". <http://www.omg.org/cgi-bin/ doc7formal/03-03-01> . 2003.

[22] J. Warmer et A. Kleppe. "The Object Constraint Language. Precise Modeling with UML". Object Technology Series, Addison-Wesley. 1998.

[23] C. Parent, S. Spaccapietra and E. Zimanyi. Conceptual Modeling for Traditional and Spatio-Temporal Applications - The MADS Approach. In Springer Verlag, 2006.