EXTENDING OBJECT ORIENTED DATABASES TO SUPPORT THE VIEWPOINT MECHANISM

Fouzia Benchikha, Mahmoud Boufaida
LIRE Laboratory, Departement of Computer Science, Mentouri university of Constantine
25000 Constantine, Algeria

Keywords: Object-oriented databases, data model, Viewpoint approach, Distributed databases.

Abstract: An important dimension in the database technology evolution is the development of advanced/sophisticated database models. In particular, the viewpoint concept receives a widespread attention. Its integration to a data model gives a flexibility for the conventional object-oriented data model and allows one to improve the modeling power of objects. On the other hand, the viewpoint concept can be used as a means to master the complexity of the current systems permitting a distributed manner to develop them. In this paper we propose a data model MVDB (Multi-Viewpoint DataBase model) that extends the object database model with the viewpoint mechanism. The viewpoint notion is used as an approach for a distributed development of a database schema, as a means for object multiple description and as a mechanism for dealing with the integrity constraint problems commonly met in distributed environment.

1 INTRODUCTION

The expansion and the distribution of databases towards the new co-operative applications of increased complexity need a new way to schema design that fulfills several requirements of data modeling. However, in complex applications, like computer aided design and engineering, it is difficult to work out a single abstraction, which is appropriate to all the participants of a same project. Each expert, according to his field of knowledge and his objectives, focuses on certain aspects of the Universe of Discourse (UoD), which are not the same ones for an another expert. Thus, it would be necessary to specify data so as to take account several points of view, while keeping to each one its specificity and allowing the sharing and the exchange of information. This perception mode of data is called the “viewpoint approach".

The viewpoint paradigm takes various significances according to its study from diverse standpoints by different fields of computer science including software engineering (Charrel, and al. 1993), knowledge representation (Dekker, 1994), database systems (Rundenteiner, 1992) (Bertino, 1992), web applications (Gergatsoulis, and al. 2001), requirements engineering (Menzies, and al. 1999) and complex systems modeling (Carn, 1992). Several terms have been assigned to this concept such as roles, aspects, perspectives, dimensions, domains and viewpoints.

In databases, we notice few works on the integration of the viewpoint concept in the data models. Most of these works consider the view mechanism. Views are external schemas that provide to the user part of the global schema, a kind of *viewpoint* on the description of its entities. In the context of our work, viewpoints accords to the same UoD several descriptions. Each description is a partial representation of data according to a given “viewpoint”. The various partial descriptions are supported by database schemas that together provide the global schema of the same real word data. The fact of achieving such an approach requires necessarily a distributed environment that permits the integration and the collaboration of a collection of databases.

In this paper, we propose extending the conventional object data model to support the viewpoint mechanism. The choice of the object oriented paradigm is primarily due to the basic concepts that it offers, in particular a great power of expression, a better reutilisability and evolution of objects. These concepts correspond to the new aspirations of the viewpoint notion. We propose a new data model MVDB (Multi-View-Database model) that uses viewpoints as an approach for a distributed development of a database schemas.

Benchikha F. and Boufaida M. (2005).
EXTENDING OBJECT ORIENTED DATABASES TO SUPPORT THE VIEWPOINT MECHANISM.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 273-278
DOI: 10.5220/0002532602730278

The paper is structured in the following way. In section 2, some related works concerning the integration of viewpoints in databases are briefly presented. Section 3 defines the MVDB structures and section 4 concludes our work.

2 RELATED WORKS

In the database field, the concept of viewpoint is mainly studied within the framework of view mechanism in object-oriented databases. Various view models are proposed such as the view model of (Abiteboul, 1991), the view model of Rundensteiner (Rundenteiner, 1992) and the view model of Bertino (Bertino, 1992). In these works views are exploited to allow different applications to see the same database according to different viewpoints. The viewpoint concept here supports external schema, which is the third level of the ANSI architecture standard upon which the construction and the use of relational database systems and later object-oriented ones are centred. Many problems are arised like how a view schema (view class) is inserted in the global schema (class hierarchy) and does an instance of a view own an identity.

Abiteboul (Abiteboul, 1991) provides a general framework for view definition. A virtual class mechanism is used for instantiating views in object-oriented databases. Here, classes for views are explicitly defined where the attributes of these classes are really methods that retrieve the information from where it is actually stored. A view can be treated as a database but it does not preserve object identity.

Rundensteiner (Rundenteiner, 1992) and Bertino (Bertino, 1992) introduce the concepts of multiview and schema view, respectively, that provide the capacity to restructure a database schema so that it meets the need of specific applications. They present support for view design by automating some tasks of the view specification process and by supporting automatic tools for enforcing the consistency of a view schema. Indeed, different views of the same object is allowed, depending on the context in which the object is considered. Here views preserve object identity but there is any relation between the various instances of the same object.

All these models consider the viewpoint as a view or external schema defined with the aim of adapting an existing structure to new needs. In our study we argue that the view and the viewpoint concepts concern respectively the exploitation step and the design one (Benchikha and al. 2001). A viewpoint must directly be related to the objects description and confers new properties to them. Our MVDB model deals with this requirement and allows the construction of distributed object databases based on viewpoints.

3 MVDB MODEL

In this section we present the basic notions and concepts of our model.

3.1 Basic notions

MVDB model is based on the object oriented paradigm intensively studied within the DBs framework. We adopt this object model as the common one for the various database schemas.

In the MVDB model a database is a multiple description of the same UoD. The database schema, shown in figure 1, is viewed as a set of viewpoint schemas (VP schemas) managed by different database systems. Each viewpoint schema represents an aspect of the data description and is held by an independent database. The viewpoint schemas construction is based on the referential one that holds basic data on the real word entities shared by all viewpoint schemas. Objects in the referential base are called global ones. Global objects have a basic description in the referential base and one or more descriptions according to viewpoints.

We point that object identity is a central notion in our approach. It is the same object described in a multiple way according to its membership in the various viewpoint schemas. However, in order to ensure the components autonomy, local objects can be created and managed locally by viewpoint databases. Local objects are objects with a single description according to one viewpoint and can't be accessed at the global level.

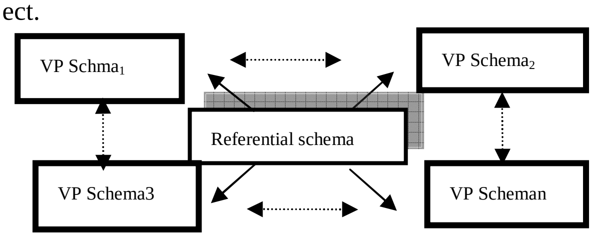

Figure 1: An example of a global schema description according to multiple viewpoints

Viewpoint databases are complementary and provide a global distributed database called multi-viewpoint database. A coherent exploitation of this global database is then recommended. This modeling approach confers a decentralized vision of a database schema, facilitates the parallel work of several designers and leads to a great power of data structuring. Generally these features are particularly needed in the great complex applications of the industrial world. As a matter of fact, the companies are at least logically distributed in offices, departments, working groups, etc. Consequently we can deduce that the data are also already distributed. Each unit in the company must possibly manage the relevant data for its operation and should be able, if necessary, to reach remote data that exist in the other units. The data in the various units are complementary and operated by collaborating users.

We illustrate the viewpoint approach through a simple modeling example. It concerns the representation of a laboratory's scientific staff (see figure 2).

We are particularly interested in the teaching and research activities concerning each member of the laboratory. Let us consider the following viewpoints: the Research viewpoint and the Teaching viewpoint. A basic Referential consists of the common information to all the viewpoints (family name, last name and age of the members). On the other hand, each viewpoint contains only information, that is relevant to it. The Research viewpoint, for example, contains, for each member of the laboratory, its research topic, its research-institution and its research-time.

3.2 Formal model

An object-oriented database (ODB), according to the database technology, contains two fundamental sorts of information: data, represented by the object's state (the database *extent*), and metadata, implemented in the database schema.

An ODB schema is organized as a class hierarchy according to the sub-classing relationship. This last provides the conditions to establish if two classes are in a generalisation/specialization. MVDB proposes an extension of the conventional object model to support the viewpoint paradigm.

Let be MVDB (Sr, VP, C, O) the specification of the data model signature, where:

- Sr is a referential schema name.
- VP is a set of viewpoint schema names,
- C is a set of class names,
- O a set of objects.

In the following, we define formally the basic concepts of the model.

Definition 1. Referential schema
The referential schema is defined as any common object schema $Sr = (Cr , \propto r)$ where:
Cr ($Cr \in C$) is the finite set of class names in the schema and $\propto r$ is the sub-classing relationship, which is a strict partial ordering among Cr.
Sr is well-formed iff:
$\forall (c, c') \in Cr \times Cr ,\ c \propto c' \Rightarrow (Ext(c) \subseteq Ext(c'))$
where Ext is the extension of a class.

The referential schema is the basic schema that describes all the entities independently from any viewpoint. A viewpoint schema is the customization of the whole or of a part of the referential schema defined as follows.

Definition 2. Viewpoint schema
Let be $S = (C, \propto)$ and $S' = (C', \propto')$ two schemas.
S' is a *projection* of S (conversely, S is an *extension* of S' on C), denoted $S' \nabla S$ (conversely $S \Delta S'$) iff: $(C' \subseteq C)$ and $(\sigma'_{|C'} = \sigma)$

A viewpoint schema Svp is an extension of a projection S' on the referential schema Sr (as depicted in figure 3). A viewpoint schema is obtained by two steps: at first, a projection operation (∇) is carried out on the referential schema to select

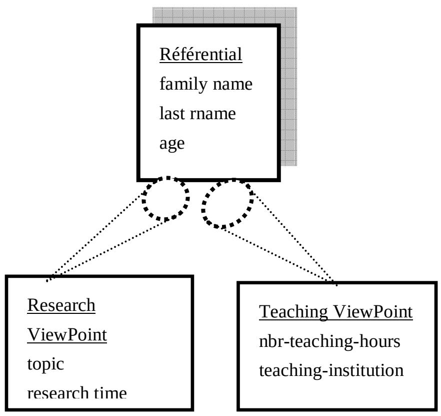

Figure 2: A multi-viewpoint modeling example

the part (the whole) of it, which will be described according to the considered viewpoint. Then, an extension operation (Δ) of the resulting schema customizes the entities description according to the viewpoint. However, in order to support independence between the various viewpoint schemas, and to keep the specificity of each one, we choose a decentralized description. For that, we benefit from the "slicing technique" used in certain approaches as described in (Bertino, 1992) (Rundensteiner, 1992). This technique consists of the distribution of the referential projected schema in the viewpoint schema.

In the following, we give an example showing the definition of the multi-viewpoint database that describes the laboratory's scientific staff schema. The laboratory-schema is the referential schema that models information about members of a given laboratory. Each member is an object stored in laboratory-base. The research-viewpoint schema refines the members description by adding new attributes. All the members are concerned with this description here. The whole of the referential schema is thus imported. The schema definition is:

Referential schema Definition
Schema laboratory-schema ;
Base laboratory-base;
Class member
  Public type tuple (
    family-name : string,
    last-name : string,
    age : integer**)**
End;
Name members = **set (**member**);**
Export-schema class member;
Export-base name members;

Viewpoint schema Definition
Viewpoint research-viewpoint **from** laboratory;
Base researchers-base;
Import-schema laboratory-schema **class** member;
Import-schema laboratory-base **name** members;
Class research **from** member
  Public type tuple (
    research-time : integer,
    research-Institution : string**)**
End.

The various schemas (referential and viewpoints) hold all the persistent objects, instances of the different schema classes. Before giving the formal definition of them, we first present the extension that is brought to the object concept.

In the context of our work, we address the two following object properties.

Property 1: An object is an instance of the referential schema and an instance of one or several viewpoint schemas.

Thus, an object has a basic description and may be described according to various viewpoints simultaneously. This is of the most broadly accepted properties of the viewpoint concept. Because an object in a viewpoint schema is seen as an instance of a viewpoint schema, it amounts to the multiple description of the objects.

Property 2: The state of an object is viewpoint oriented.

It means that the state of an object may vary depending on the viewpoints in which it is being described. This seems to suggest that each description of an object according to a viewpoint should be viewed as a separate instance of it. This property that allows the object multiple instanciation suits together with the first one.
According to the afore mentioned properties, we integrate a new concept, that is the *object referent.* We distinguish a local object referent from a global one.

Definition 3. Local object referent
A local object referent, denoted by r_l, is the identification of the object in a viewpoint (local) database such that :
  $r_l = (vp, o_l)$ where :
  – vp is the viewpoint schema name,
  – o_l is a viewpoint identification (OID).
An object in a viewpoint database becomes a pair (r_l, v_l) where v_l is its local state value.

Definition 4. Global object referent
A global object referent, denoted by r_g, represents the identification of the object in the multi-viewpoint database such that :

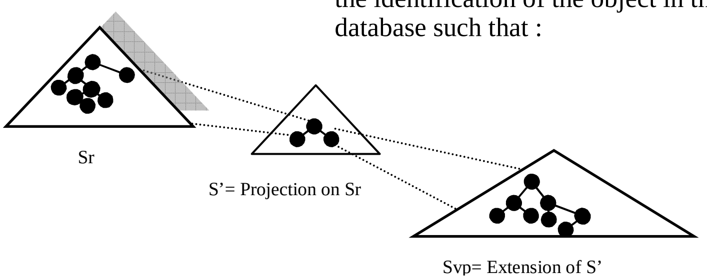

Figure 3: Viewpoint schema = Projection + Extension of the referential schema

$r_g = (o_g, , L(r_l))$ where :

- o_g is the referential OID of the object,
- L is the list of its viewpoint identifications that is : $L(r_l) = U_{vp \in VP} (r_l)$.

An object becomes a pair (r_g , v_g) where v_g is its global state value.

The objects constitute the associated bases of the referential and the viewpoint schemas that are defined in the following.

Definition 5. Viewpoint base

Let O_{vp} be the finite set of the objects identifier of a viewpoint schema S_{vp}.

A viewpoint base, denoted Bvp, is a schema state specified commonly as a tuple (π_{vp}, O_{vp}, $\surd_{vp}$) where :

- $\pi_{vp} : C_{vp} \rightarrow O_{vp}$ is the function that associates to each class $c \in C_{vp}$ its object identifiers,
- O_{vp} is the set of the objects in B_{vp} : $O_{vp} = U_{c \in Cvp} \{\pi_{vp} (c)\}$,
- $\surd_{vp}$ is the function that associates a value to each object of B_{vp}, such as : $\forall c \in C_{vp}, \forall o \in \pi_{vp} (c) , \surd_{vp}(o) \in \sigma_{vp}(c)$.

Definition 6. Referential base

Let R_r be the finite set of the objects referent of the referential schema S_r and VP the finite set of the viewpoints schema names.

A referential base, denoted Br, is a schema state specified as a tuple (π_r, O_r, $\surd_r$) where :

- $\pi_r : C_r \rightarrow R_r$ is the function that associates to each class $c \in C_r$ the objects referent,
- O_r is the set of the objects in Br : $O_r = U_{c \in Cr} \{(\pi_r (c), U_{vp \in VP} (rl))\}$,
- $\surd_r$ is the function that associates a global value to each object of B_r, such us : $\forall c \in C_r, \forall o \in \pi_r (c) , \surd_r(o) \in \sigma_r(c)$. where Br describes the UoD entities.

Referential base and viewpoint bases constitute the multi-viewpoint base (extent) of the multi-viewpoint schema. They give the complete description of objects according to multiple viewpoints. Lus us illustrate this by the following example.

The associated bases of the referential and the viewpoint schemas presented in the above example are populated with objects. Each object has an instance in the 'laboratory-base' (referential base) and can possibly be instantiated in the 'research-base'. Instances are collected under the root of persistence of each schema as presented in follows.

Bases

laboratory-base ((oid1, {Benali, Mohamed, 40 }) ,
(research-vp , oidvp1))
(oid2, {Bencharif, Ali, 55 }, nil)
researchers-base (oidvp1, { Benali, Mohamed, 40 , 12, 'Constantine-Univer' })

In the following we focus a bit more on an essential and complementary functionality to every data model: the coherence. This one is directly related to the integrity constraints study.

3.3 The integrity constraints in MVDB

Unlike the traditional approach (mono viewpoint) where the integrity constraints are defined on the global schema, we distinguish in the viewpoint approach two types of constraints.

- Local constraints: they contribute to ensure the local coherence of the entities in a viewpoint database and independently of the other bases.
- Global constraints: they contribute to ensure the global description coherence of the entities according to several viewpoints.

If the local constraints are apprehended, it is difficult to take into account the global constraints. Classically, the principal conflicts met in federated databases are the names, the semantic and the structural conflicts. In our work, these latter can be solved by the viewpoint mechanism.

- Naming conflicts: traditionally, the solving of this type of conflicts is done by assertions specifying the synonyms and the homonyms. In our context, the existence of the referential solves any conflict coming from a problem of synonymy. Thus, all the common properties are described by the referential schema. On the other hand, a conflict coming from homonyms is solved by the viewpoint mechanism itself. As a matter of fact, two distinct homonymous constructions can be differentiated by prefixing them, for example, by the name of the partial schema.
- Semantic and structural conflicts: they are of a weak or a lack, in a database schema designed according to various viewpoints. Nevertheless, each partial schema describes an aspect of the data semantically different from the other descriptions. In addition, the referential permits a representation, and by the same way an unified structure of the real world entities that will have different descriptions according to different viewpoints .

However, within the framework of MVDB, we distinguish other types of conflicts. These ones permit to guarantee the compatibility and the coordination between the different object descriptions in the system. Let us consider the following cases.

- Mutual exclusion between partial DBs: when the description of the entities by a partial schema compromises their description by another partial schema.

Example: any temporary teacher does not have the right to acquire a teaching activity, and cannot have a description according to this viewpoint.

- Interdependency between partial DBs: when the partial schemas contain linked properties.

Example: any teacher whose the research time exceeds twenty hours a week must reduce his official teaching time by 40 percent.

- Referential integrity between partial DBs: when the creation (possibly deleting) of a database entity requires a preliminary creation (possibly a deleting) of one (or many) entity (ies) of another databases.

Example: any teacher must be a member of a research group. This implies that the creation of any object instance according to the teaching viewpoint must generate the creation of the same object instance in the research viewpoint.

A multi-viewpoint base is coherent (here we speak about coherence with respect to the semantics of the applications and not about the implicit coherence induced by the model) if the following conditions are satisfied:

1. The referential base is locally coherent with respect to its schema, i.e. the object set checks the local constraints of the referential schema.
2. Each viewpoint base is coherent with the corresponding viewpoint schema, i.e. each viewpoint objects check the viewpoint schema constraints.
3. The gathering of the various bases is coherent, i.e. all the global constraints are satisfied.

4 CONCLUSION

In this paper we have sketched how the object oriented database model can be extended to support the viewpoint mechanism. The ability to express multiple viewpoints allows for database construction versatility, richer expression of data semantics and permits a distributed manner for design, which is not captured by the current generation of object oriented database systems. Indeed, it could be interesting to study the behaviour aspects of the objects that can be very useful for the constraint management that we consider. In addition, it would be interesting to develop an expression language for the specification of the integrity constraints and extend a query language like OQL, for dealing with the multi-viewpoint aspect of the objects. These last can then be a query according to one or several viewpoints.

REFERENCES

Abiteboul, S. & Bonner, A., 1991. Objects and Views. *In ACM SIGMOD* (pp. 238-247).

Benchikha, F., Boufaida, M. & Seinturier, L., 2001. The integration of the Viewpoint Mechanism in Federated Databases. *In ACM, SAC'01* (pp. 280-284). Las Vegas.

Bertino, E., 1992. A View Mechanism for Object-Oriented Databases. *In the 3rd International Conference on EDTB'92* (pp. 136-151). Australia.

Carn, N., 1992. Représentation Orientée Objet de Système Opérationnel avec application au domaine spacial. *Thesis*, Toulouse, France.

Charrel, P. J., Galaretta, D., Hanachi, C., & Rothenburger, B., 1993. Multiple Viewpoints for the Development of Complex Software. *In IEEE International Conference on Systems*, Man and Cybernetics (pp. 556-561).

Dekker, L., 1994. FROME: Représentation Multiple et Classification d'Objets avec Points de Vues". *Doctorat thesis*, University of sciences and technologies, Lile, France.

Gergatsoulis, M., Stavrakas, Y., Karteris, D., Mouzaki A., & Sterpis, D., 2001. A Web-Based System for Handling Multidimentional Information through MXML. *In Lecture Notes In Computer Science* (LNCS 2151) (pp. 352-365), Springer-Verlag.

Menzies, T., Easterbrook, S., Nuseibeh, B., & Waugh, S., 1999. An Empirical investigation of multiple viewpoint reasoning in requirements engineering. *In the fourth International Symposium on Requirements Engineering* (RE'99), Limerick, Ireland.

Rundensteiner, E., 1992. Multiview: a Methodology for supporting Multiple Views in Object-oriented Databases. *In the 18th VLDB Conference*. (pp. 187-198), Canada.

Wieczerzycki, W., 1999. Database model for web-based cooperative applications. *In CIKM'99* (pp. 131-138) USA.