Approche pour l'intégration du temps dans la conception de bases de données objet

M.Yefsah et I. Rassoul

Institut d'informatique. Université Mouloud Mammeri. Tizi-Ouzou
B.P 531 RP 15000 Tizi Ouzou
Tel : 03 21 51 08
Fax : (03) 21- 77- 78

Résumé

Nous proposons dans cet article une approche pour l'intégration du temps dans la conception de bases de données objet. L'analyse des nombreux travaux sur ce sujet montre que les solutions apportées à ce problème sont conçues comme un mécanisme d'implémentation de versions d'objets avec un estampillage temporel par attribut ou par version d'objet. Cette mise en œuvre s'apparente à une approche procédurale, qui correspond à prendre en charge la gestion des versions au niveau programme. Pour remédier à cette contrainte, nous proposons une approche déclarative qui consiste à exprimer au niveau du modèle toute la réalité temporelle en utilisant le concept d'association temporelle.

Mots clés : Base de données objet, base de données temporelles, modèles de données temporelles, Association temporelle.

Abstract

In this paper, we propose an approach for the integration of the time constraint in the conception of object Databases. The analysis of the different efforts for this matter reveals that the solutions proposed for this problem can be regarded as an implementation mechanism of versions of objects with a temporal number (« estampille ») per attribute or per version of an object. This way of conception falls within the procedural approach, which consists of taking the management of the versions at the program level. To respond to this constraint, we propose a declarative approach to describe the temporal reality at the model level by the use of temporal association concept.

Key words : Object Databases, temporal Databases, Temporal Data models, Temporal association.

1. INTRODUCTION

La prise en compte explicite du temps dans le domaine des systèmes d'information en général et des bases de données en particulier a constitué depuis une vingtaine d'années un sujet qui a concentré l'attention de plusieurs chercheurs. Les principaux travaux de représentation explicite des aspects temporels dans les Bases de données ont pour cadre le modèle relationnel, [CLIF 85], [CLIF 87], [CLIF 93], [GADI 88a], [NAVA 90] et dans une moindre mesure les modèles sémantiques [KLOP 83], [SEGE 87], [ELMA 90],[DUBO 86] et l'orienté Objet. La représentation du temps dans les systèmes de gestion de base de données, s'est concentrée autour de l'extension du système relationnel. Pratiquement, cela s'est fait par l'association du temps au niveau des tuples [BENZ 82], [NAVA 89], [SARD 90], [SNOD 87], ou par l'ajout du temps au niveau du domaine d'attribut [CLIF 87], [GADI 88a], [LORE 88], [MCKE 91], [TANS 86]. Ces travaux ont mis en évidence un ensemble de concepts et de notions temporelles qui constitue un apport important pour mieux cerner le cadre de conception des bases de données temporelles. La définition d'un glossaire de concepts [JENS 94], a permit de stabiliser la terminologie qui a connu en ses débuts une situation chaotique.

Le modèle relationnel ne permet pas une représentation satisfaisante des propriétés structurelles et comportementales du monde réel. Les limites du modèle relationnel ont poussées à chercher une autre génération de modèles de données, particulièrement orienté objet.

Les modèles de données Orienté objet offrent plusieurs propriétés telles que l'héritage, l'identité de l'objet, encapsulation, polymorphisme, et le mécanisme type/classe, qui permettent de modéliser la réalité avec plus de fidélité. Un schéma de base de données objet, peut être vu comme un réseau de classes d'objets interconnectés par un ensemble de liens.

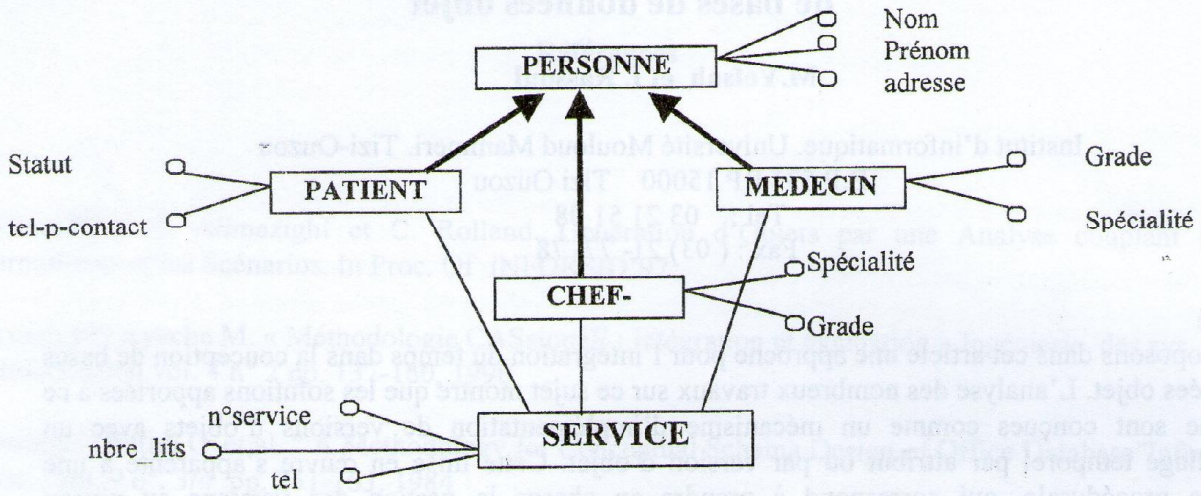


Figure 1 : Schéma graphique de la base de données PATIENT

Le statut du patient, peut prendre plusieurs valeurs parmi : {en observation, en traitement, sortant, sous surveillance externe, ...}.
Du schéma précédent, nous constatons qu'il est facile de représenter les patients actuels d'un service, les médecins qui sont en exercice, le chef de service actuel, mais il est difficile de représenter et gérer l'évolution du service à travers l'évolution du personnel et des responsables qui se sont succédés. Prendre en compte cette dimension historique nécessite non seulement d'associer le temps aux propriétés qui en dépendent (attributs ou objets) mais aussi les liens qui existent entre celles-ci.
Dans le contexte des bases de données, deux dimensions temps sont en général considérées, le temps de validité et le temps de transaction [SNOD 86]. Ces deux dimensions sont orthogonales. Les bases de données intégrant le temps de validité sont appelées base de données historiques, celles intégrant le temps de transaction sont appelées base de données de reprise. Les bases intégrant les deux dimensions sont, quant à elles appelées les bases de données bitemporelles. L'intérêt d'une telle distinction réside dans la possibilité de pouvoir modéliser des traitements de type rétroactifs.
Par exemple, dans la base de données PATIENT, l'enregistrement de la disponibilité d'un lit dans un service donné deux semaines avant sa date d'effet (vacance effective) demande un traitement rétroactif nécessitant les connaissances à la fois de la date d'enregistrement et de la date d'effet de la libération du lit en question. Une étude détaillée de la relation entre les deux dimensions peut être trouvée dans [JENS 92c].
Nous présentons dans la section 2 les principaux travaux existants, la section 3 présente un exemple afin de mettre en évidence les inconvénients de l'approche par mécanisme de chaînage. La section 4 revient sur cette approche pour montrer la nécessité d'introduire l'approche proposée. La section 5 présente l'approche déclarative par l'utilisation de l'association temporelle et quelques types d'associations et finalement on présente quelques perspectives en section 6.

2. Travaux existants

Plusieurs modèles objets temporels ont été proposés ces dernières années, OSAM/T [SUCH 91], TMAD [KAFE 92], OODAPLEX [DAYA 92], TIGUKAT [GORA 93], TOODM [ROSE 91]. On trouvera dans [SNOD 95a] un panorama de diverses propositions d'extensions temporelles des modèles à objets. Selon les systèmes, les historiques sont définis au niveau des objets [SUCH91] [KAFE 92] ou des attributs[ROSE91] ou indifféremment au niveau des objets ou des attributs [GORA 93], [DAYA 92].
Dans le cas du versionnement des attributs, chaque attribut a pour valeur une liste de paires ***<valeur, temps>*** ou temps indique le temps de validité de cette valeur pour l'attribut. Le principe ***d'homogénéité*** temporelle impose que tous les attributs d'un objet aient la même période de vie. Cette propriété garantit l'absence de valeurs nulles ou indéfinies

Dans le cas du versionnement globale des objets, chaque objet possède des attributs supplémentaires correspondants. Le cadre de conception de base de données temporelles orienté objet défini par Iqbal [IQBA98] est constitué de quatre types :

- Le type calendrier désigne la solution apportée pour l'implémentation de la dimension temporelle et les granularité offerte par le modèle.
- Le type structure temporelle, fournit l'ontologie sous-jacente et les domaines de temps.
- Le type d'ordre temporel est relatif au modèle de temps, fournit un ordre au temps.
- Le type historique temporel est relatif à l'association du temps aux événements, données et activités du domaine.

Dans le tableau qui suit, nous avons regroupé les caractéristiques des modèles temporels orienté objet selon le cadre de conception présenté.

Modèle	**Structure temporelle**			**Calendrier**	**Ordre temporel**	**Historique**
	Domaine	**Définition**	**Primitives**			
TOODM	Continu	Information Déterminé	Absolu Relatif	Grégorien	Linéaire total	Validité Transaction Evénement
OSAM/T	Discret	Déterminé	Absolu	N/D	Linéaire	Validité
TMAD	Discret	Déterminé	Absolu	Grégorien	Linéaire	Validité Transaction
TGUKAT	Discret	Déterminé	Absolu	N/D	Linéaire	Validité
OODAPLEX	Discret	Déterminé	Absolu	N/D	Linéaire	Validité

Structure temporelle. Notons que la plupart des modèles supportent une structure temporelle simple, qui consiste en un ensemble de primitives absolues sur une ligne de temps discrète et manipulant l'information déterminée. La structure temporelle de TOODM se distingue par le fait qu'il permet de gérer le temps relatif, et il est le seul à supporter le domaine temporel continu.
Calendrier. Les modèles de primitives temporelles dans OSAM/T, OODAPLEX et TIGUKAT sont représentés en utilisant simplement les entiers naturels. Ces modèles n'offrent aucun système de datation calendaire avec de multiples granularités. Les autres modèles offrent un seul schéma de représentation pour les primitives temporelles, à savoir le calendrier grégorien.
Ordre temporel. Tous les modèles présentés supportent l'ordre temporel linéaire.
Historiques. Le temps de validité est considéré comme la dimension temporelle classique que tout modèle temporel doit supporter. Le temps de transaction est pris en compte par certains modèles afin d'affiner les traitements temporels et faciliter le contrôle de cohérence des bases de données. Le temps événement, bien qu'il constitue un moyen formidable pour capter une plus grande sémantique du comportement du monde réel à modéliser, n'est considéré que par un nombre très limité de modèles.

3. Exemple

Considérons les informations suivantes :

Le service Médecine interne a été crée en 1980 mais ne disposait d'aucun médecin.
[1985, 1989] le médecin M1 était seul dans le service. [1990, 1997] M1 et M2 étaient ensemble au service. [1998, Now] nous n'avons pas l'information.
[1985, 1988] le médecin M1 est généraliste,[1989, 1992] il est résident, [1993, Now] il est assistant.
[1990,1993] le médecin M2 est généraliste, [1994, Now] il est résident.
[1985, 1997] M1 est affecté au service médecine interne,[1998, Now] service de M1 inconnu
[1990, 1997] M2 est affecté au service médecine interne,[1998, Now] service de M2 inconnu.

3.1 Estampillage par attribut (version d'attribut)

Temps validité	Médecin	Grade
[1985, 1988]	M1	Gen
[1989, 1992]	M1	Res
[1993, Now]	M1	Ass
[1990,1993]	M2	Gen
[1994, Now]	M2	Res

Temps validité	Service	Médecin
[1980,1984]	med-int	-
[1985,1989	med-int	M1
[1990,1997]	med-int	M1,M2
[1998,Now]	Med-Int	-

Temps validité	Médecin	Service
[1985, 1997]	M1	Med-int
[1998,Now]	M1	-
[1990,1997]	M2	méd-Int
[1998,Now]	M2	-

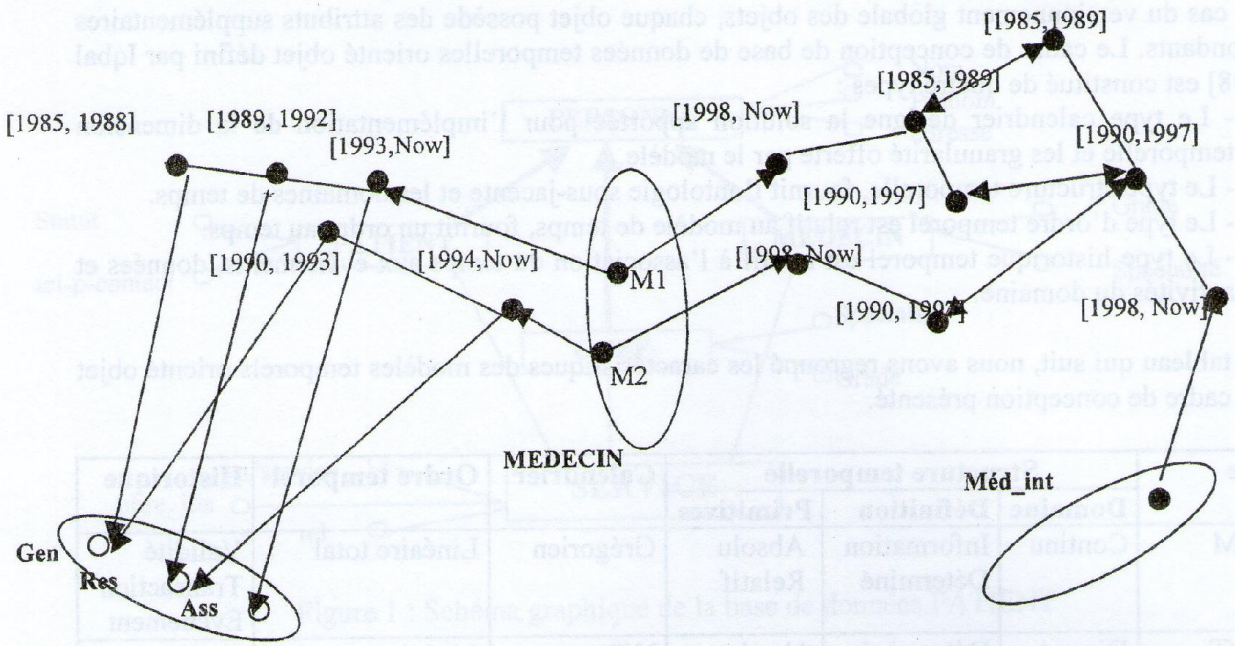
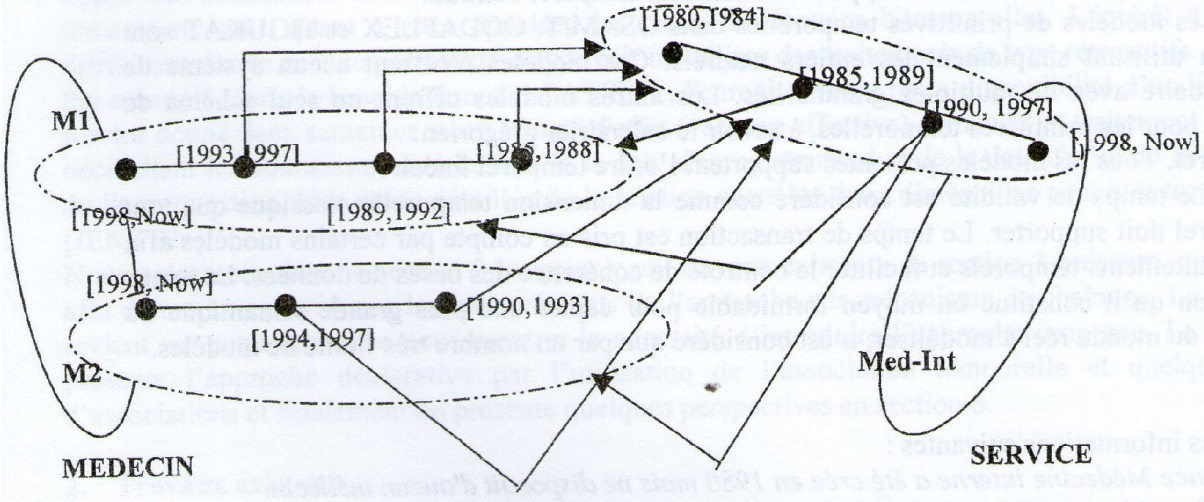


Figure 2 : schéma au niveau des occurrences selon un estampillage par attribut

3.2 Estampillage par objet (version d'objet)

Temps validité	Service	Médecin
[1980,1984]	Med-Int	-
[1985,1989	Med-Int	M1
[1990,1997]	Med-Int	M1,M2
[1998,Now]	Med-Int	-

Temps validité	Médecin	Grade	service
[1985,1988]	M1	Gen	Med-Int
[1989,1992]	M1	Res	Med-Int
[1993,1997]	M1	Ass	Med-Int
[1998,Now]	M1	Ass	-
[1990,1993]	M1	Gen	Med-Int
[1994,1997]	M2	Res	Med-Int
[1998,Now]	M2	Res	-



Légende : Instance d'objet estampillé selon le temps de validité

Figure 3 : Schéma au niveau des occurrences selon un estampillage par objet

Comme on vient de le voir, il est possible de répondre d'une manière satisfaisante au problème posé en section 1, par l'adjonction du temps au modèle orienté objet soit au niveau attribut (exemple 3.1) ou de l'objet (exemple 3.2). Ainsi, dans les schémas au niveau des occurrences (figure 4 et 5), nous avons représenté l'évolution du service médecine interne à travers l'évolution de son personnel médical, ainsi que l'évolution de la qualification du personnel médical.

3.2.2 Instructions

The SIMD instruction (macro instruction) word consists of the following subfields: opcode, address mode identification and register identification of first operand, address mode identification, register identification of second operand and a constante.

We represent the set of instructions by an abstract data type namely *Instruction* that exports the sorts *inst, opcode, address-mode, reg-ident* with the operators,

- *code(I)* delivering the opcode of the instruction *I* (*is-equal, test-mark, set-mark, move* and *init*) ,
- *mod1(I)* and *mod2(I)* delivring the address mode identification respectively of the first and second operand (immediate, direct or indirect address mode),
- *reg1(I)* and *reg2(I)* delivring the register identification respectively of first and second operand (*rtoken, rleft, rright* or *rflag*).
- cst(I) delivring a constant if the address mode is direct.

```
type Instruction is BIT_ARRAY
sorts inst, opcode, address_mode, reg_ident
opns init, is-equal, test-mark, set-mark, move : → opcode
     rtoken, rflag, rleft, rright : → reg-ident
     mod1, mod2 : inst →addres_mode
     code : inst → opcode
     immediate, direct, indirect → addres_mode
     reg1, reg2 : inst → reg-ident
     cst : inst → Bit_array
endtype
```

3.3 Formal specification of a tile

We represent a tile by a process called *Tile*. The instruction bus is sepecified by the interaction point *Ins*, the data buses (north, east, south and west) connecting a tile with other adjacents tiles are specified respectively by the interaction points *DN, DE, DS* and *DW* and both request and acknowledgement buses are reprensented by the interaction points *RN, RE, RS* and *RW*. The address of tile in the ensemble is represented by the variables *line* and *column*. The term *hide* is used to indicate that *Con0, Con1, Con2 and Con3* are specified as hidden interactions points. These interaction points are not observable from the environment of *Tile* process.The tile structure is expressed in term of four cells and a management resource unit (as it is described in section 2.1). Then, it is specified in term of 4 instances of *Cell* process and an instance of *Management-Resource* process. The process *Cell* has the same interaction points that process *Tile* which are: *Ins, DN, DE, DS, DW, RN, RE, RS* and *RW*. In the first instance of Cell process, *Con0* modelizes the input of connecting signal received from *Management_ Resource* process. The number 0 represent the address of the first instance of *Cell*. All cells of a tile work in parallel and can interchange data through data buses. This behaviour is expressed by using the selective parallel composition operator */ [DN, DE, DS, DW]/* between all instances of Cell process. The management resource can send to each cell a connecting signal (Con0,..Con3). The operator */ [Con0, Con1, Con2, Con3]/* is used to synchronize each instance of process Cell on the approprate interaction point.

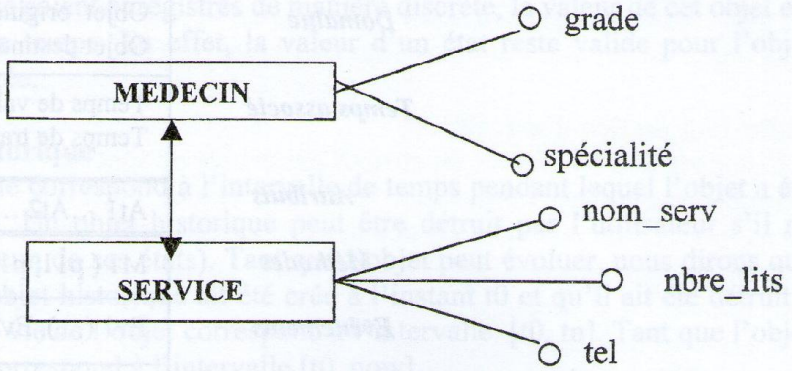
The formal specification of tile in LOTOS is the following:

```
process Tile[Ins, DN, DE, DS, DW, RN, RE, RS, RW] (line, column: Nat ): noexit:=
hide Con0,Con1, Con2, Con3 in
( Cell[Ins, DN, DE, DS, DW, RN, RE, RS, RW, Con0] (0)
            |[DN, DE, DS, DW]|
  Cell[Ins, DN, DE, DS, DW, RN, RE, RS, RW, Con1] (1)
            |[DN, DE, DS, DW]|
  Cell[Ins, DN, DE, DS, DW, RN, RE, RS, RW, Con2] (2)
            |[DN, DE, DS, DW]|
  Cell[Ins, DN, DE, DS, DW, RN, RE, RS, RW, Con3] ) (3)
            |[Con0, Con1, Con2, Con3]|
Management-Resource [RN, RE, RS, RW, Con0, Con1, Con2, Con3]
endproc (*Tile*)
```

4. Discussion

En réalité, la solution apportée dans l'exemple précédent est conçue comme un mécanisme d'implémentation de versions d'objets avec un estampillage temporel par attribut ou par version d'objet. Cette gestion ne s'appuie sur aucun modèle expliquant les raisons qui sous-tendent le versionnement des objets. En l'absence d'un tel modèle, la création des versions est une opération dont les utilisateurs ont nécessairement la charge. Cette mise en œuvre de mécanisme d'implémentation de versions d'objet s'apparente à une approche procédurale, qui correspond à prendre en charge la gestion des versions au niveau programme ou du moins une partie de celle-ci.

Reprenant l'exemple précédent :



Dès lors que nous ajoutons le temps au modèle, le lien entre l'objet MEDECIN et l'objet SERVICE prend une signification plus spécifique, qui permet de connaître l'évolution du service et l'historique du médecin tout en sauvegardant la base de données dans un état cohérent. Or, le mécanisme de gestion de versions d'objet ne permet pas de décrire toute la sémantique des objets et de leurs liens, représentée par les propriétés correspondantes et leurs contraintes d'intégrité.
Par exemple, si on prend la contrainte sur médecin qui stipule « qu'un médecin ne peut être rattaché à deux services différends en même temps », mais il peut passer d'un service à un autre. Cette contrainte s'exprime sur le lien qui existe entre l'objet médecin et l'objet service. Et si nous voulons l'exprimer d'une manière déclarative (au niveau du schéma et non du programme) nous avons besoin d'exprimer le lien entre les objets considérés sous forme d'objet spécifique. Il s'agit d'un lien de type «associationN » entre deux objets concrets et l'objet abstrait TEMPS.

5. Proposition d'implémentation du temps dans les modèles orientés objet.

Dans l'exemple de la figure 6, l'objet Ali est une instance du type d'objet CHEF-SERVICE et l'objet Médecine-interne est une instance du type SERVICE. L'association représente que Ali est le chef de service du service Médecine-interne pendant la période p.

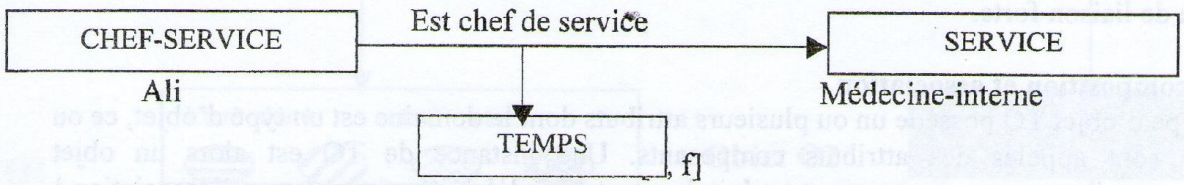


Figure 4 : exemple d'association entre deux instances d'objets

Comme il est indiqué sur la figure, l'association est ternaire. Elle met en relation deux objets en évolution en fonction du temps. Cette association permet de représenter le lien sémantique qui existe entre deux objets. Au nivcau implémentation, une association est considérée comme un objet.
Les associations servent à modéliser des relations entre deux objets. Les associations sont dirigées. Une association est établie entre un objet origine et un objet destination. L'existence de l'association est liée à l'existence de son origine et de sa destination.
Une association est une instance d'un type qui définit sa structure et son comportement.

5.1 Type d'association

Comme pour un objet, la structure et le comportement d'une association sont donnés par son type. Dans un type d'association, on trouve comme pour un type d'objet, des définitions d'attributs, de méthodes (opérations), d'événements et des contraintes. On trouve également le domaine de l'association qui détermine les types d'objets composant l'association. L'attribut origine indique le type d'objets à partir desquels l'association peut être établie, et l'attribut destination indique le type des objets qui peuvent être destination de l'association. Ce lien de type " association" peut être modélisé sous forme d'objet de type association temporelle.

Nom de l'association	Association EST CHEF DE SERVICE DE
Domaine	Objet origine : CHEFSERVICE Objet destination : SERVICE
Temps associé	Temps de validité : [début, fin] Temps de transaction : [début, fin]
Attributs	At1 ...At2 ...
Méthodes	M1 (p11, p12,...) , M2 (p21, p22, ...) ...
Evénements	Ev1 (...), Ev2 (...), ...
Contraintes d'intégrité	C1 : ..., C2 :..., ...

Figure 5 : Schéma Type objet association

⇨ Objet origine et objet destination, définissent les objets à associer en question.

⇨ Temps de validité est un intervalle dont les bornes sont le temps début de validité et temps fin de validité. Il correspond au temps de validité de l'occurrence de l'association.

⇨ Exemple de Contrainte temporelle : "un médecin ne peut être affecté à deux services différents pendant le même intervalle de temps"

5.2 Sémantique des associations

Nous pouvons définir plusieurs sous-types d'associations en fonction des types de relation et des besoins. Par exemple, la relation entre médecin et service, exprime le fait qu'un médecin soit toujours lié à un et un seul service. Ce lien signifie qu'un médecin ne peut être affecté à deux services différents en même temps d'une part et qu'il doit être affecté à au moins un service, ce qui veut dire que si on supprime service on doit supprimer automatiquement tous les médecins qui lui sont associés. Cette association possède une sémantique spécifique qui peut se résumer par :
Un objet ne peut être destination (en même temps) de deux associations. Si l'objet origine de l'association est détruit, la destination est aussi détruite. Ainsi, on vient de définir un sous-types d'association de liaison forte.

5.3 Lien de composition et association

Lorsqu'un type d'objet TO possède un ou plusieurs attributs dont le domaine est un type d'objet, ce ou ces attributs sont appelés des attributs composants. Une instance de TO est alors un objet composite. Les attributs composants seront traduit dans notre modèle par un sous-types d'association à liaison forte.

5.4 Objet historique

La notion d'objet historique est introduite pour la prise en compte de l'évolution d'un objet. La valeur d'un objet historique est une séquence d'états. Un état constitue la valeur de l'objet historique à un instant donné. Les objets historiques et les états sont représentés par des objets. Ainsi, la valeur d'un objet historique OH est une séquence d'objets d'un type T donné.

OH = [oid, {état0, état1, ..., étatn}]

Oid est l'identificateur de l'objet historique et {état0, état1, ..., étatn} est sa valeur qui représente une séquence d'états . Chaque état de l'objet historique est un objet.

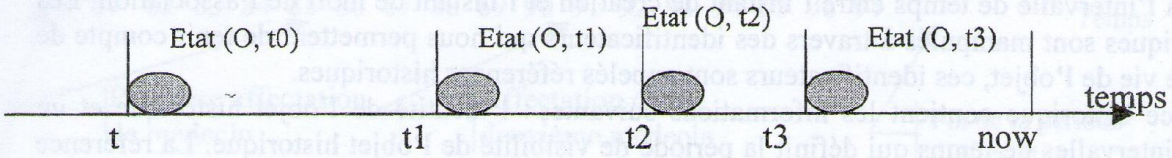


Figure 6 : Etats de l'objet historique O

Bien que les états d'un objet historique soient enregistrés de manière discrète, la valeur de cet objet est interprétée de façon continue dans le temps. En effet, la valeur d'un état reste valide pour l'objet historique tant qu'elle n'est pas changée.

5.5 La période de vie d'un objet historique

La période de vie d'un objet historique correspond à l'intervalle de temps pendant lequel l'objet a été « vivant » dans la base de données. Un objet historique peut être détruit par l'utilisateur s'il ne participe à aucune association (ni aucun de ses états). Tant que l'objet peut évoluer, nous dirons que l'objet est vivant. Supposons qu'un objet historique ait été crée à l'instant t0 et qu'il ait été détruit à l'instant tn, dans ce cas, la période de vie de l'objet correspond à l'intervalle [t0, tn]. Tant que l'objet n'est pas « mort », sa période de vie correspond à l'intervalle [t0, now].

5.6 Association et évolution d'objets

Dans une base de données historique, il est non seulement important de pouvoir représenter et gérer l'évolution des entités dans le temps mais aussi l'historique des relations entre ces entités. Plusieurs type de relations peuvent exister entre des objets historiques ou des états :

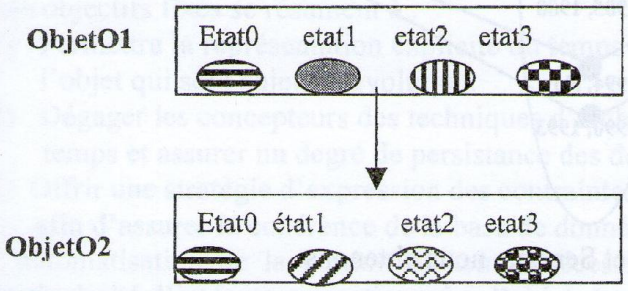


Figure 6 : Relation entre deux états d'objets

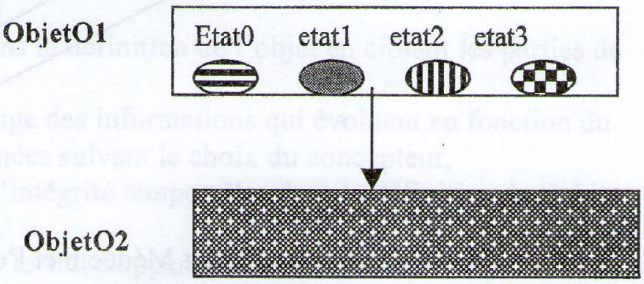


Figure 6:Relation entre un état de l'objet origine et l'objet destination

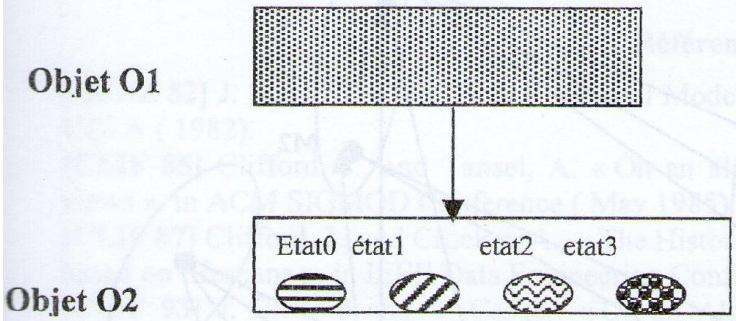


Figure 6 : Relation entre l'objet origine et un état de l'objet destination

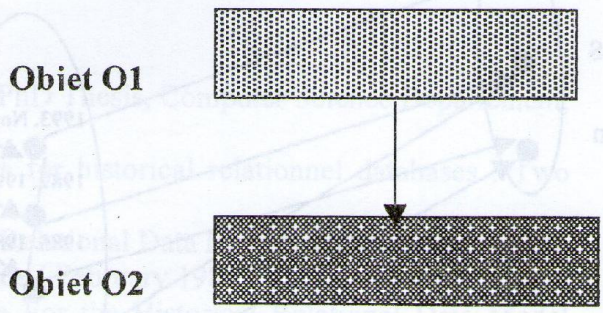


Figure 6 : Relation entre deux objets

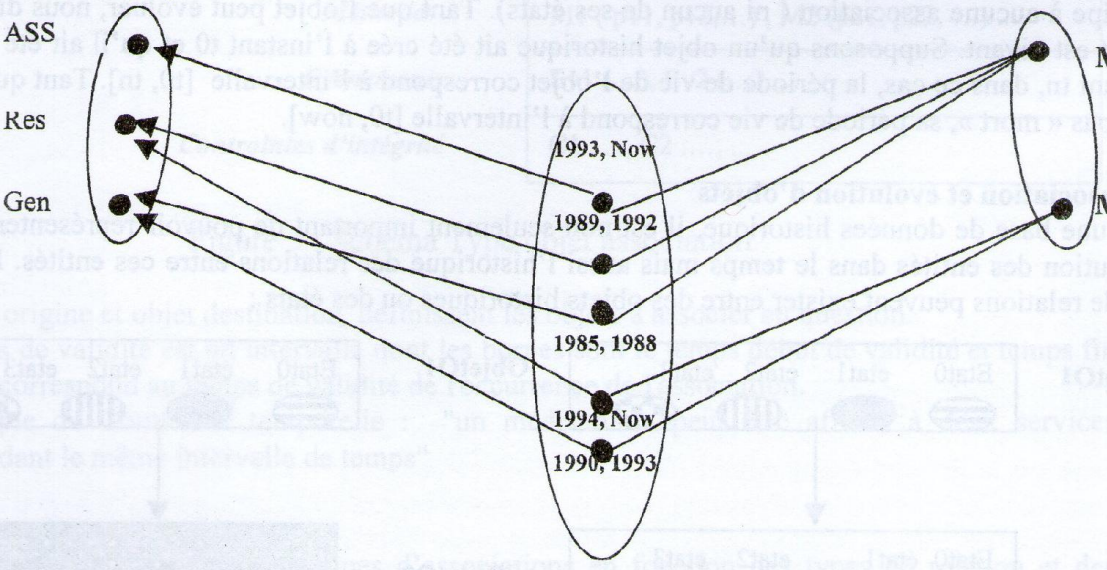
Une association impliquant un objet historique signifie que tous les états constituant l'objet historique sont eux aussi impliqués dans l'association. C'est-à-dire que l'association est partagée par tous les états de l'objet historique. Si par exemple, un objet historique est destinataire d'une association, alors chacun de ses états est aussi destination de l'association.

5.7 Associations historiques

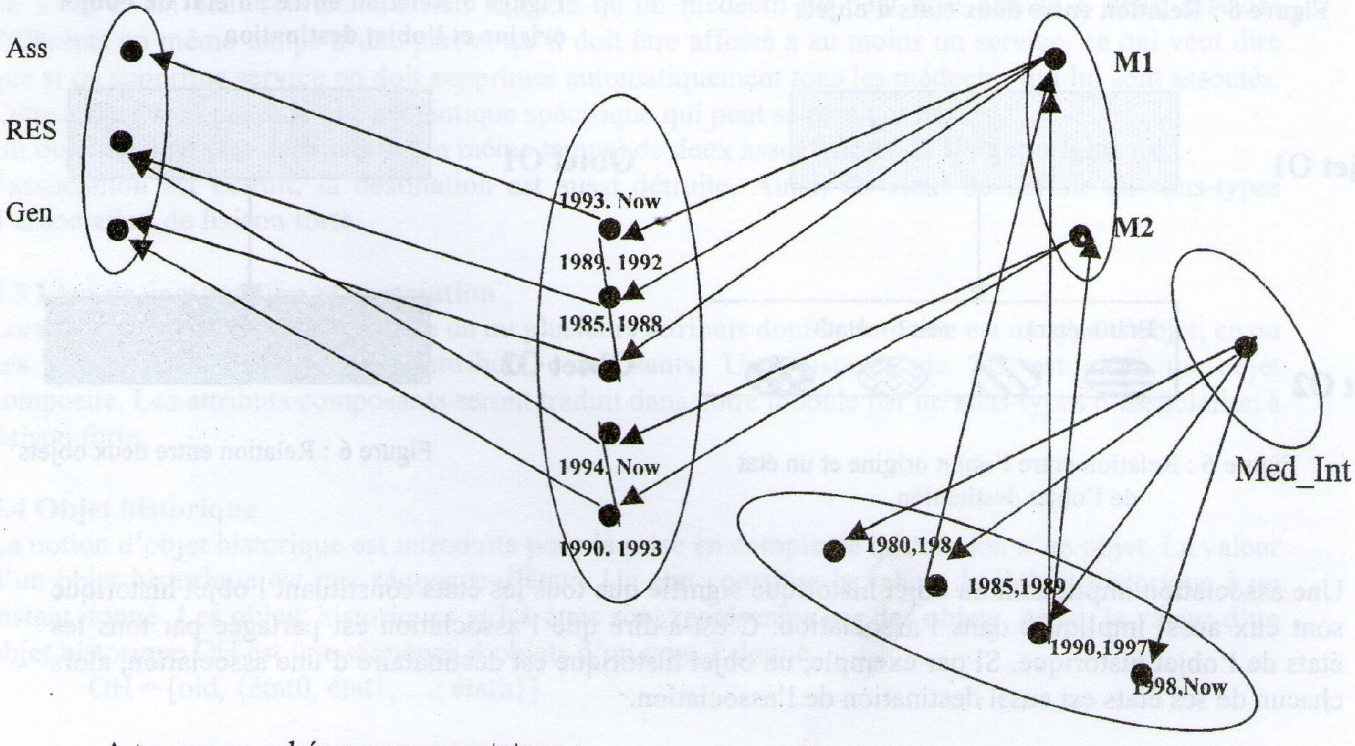
Les associations historiques sont établies entre objets historiques. Elles ont une période de validité qui correspond à l'intervalle de temps entre l'instant de création et l'instant de mort de l'association. Les objets historiques sont manipulés à travers des identificateurs qui nous permettent de tenir compte de la période de vie de l'objet, ces identificateurs sont appelés références historiques.
Une référence historique contient les informations suivantes : l'identité de l'objet historique et un ensemble d'intervalles de temps qui définit la période de visibilité de l'objet historique. La référence historique permet de restreindre la visibilité d'un objet historique à certains intervalles de temps inclus dans sa période de vie. C'est à travers des références historiques qu'est maintenue la contrainte d'existence temporelle imposée par une association historique. La notion de référence historique permet d'avoir une vue partielle de l'objet. Plusieurs références historiques peuvent être définies pour le même objet, différentes vues partielles de l'objet peuvent être obtenues simultanément.

5.8 Utilisation du lien d'association

Pour chaque attribut temporel (un attribut qui évolue en fonction du temps) d'un objet est crée automatiquement un lien d'association. Exemple, l'objet MEDECIN possède un attribut "grade" qui évolue en fonction du temps. Nous obtenons au niveau des occurrences :

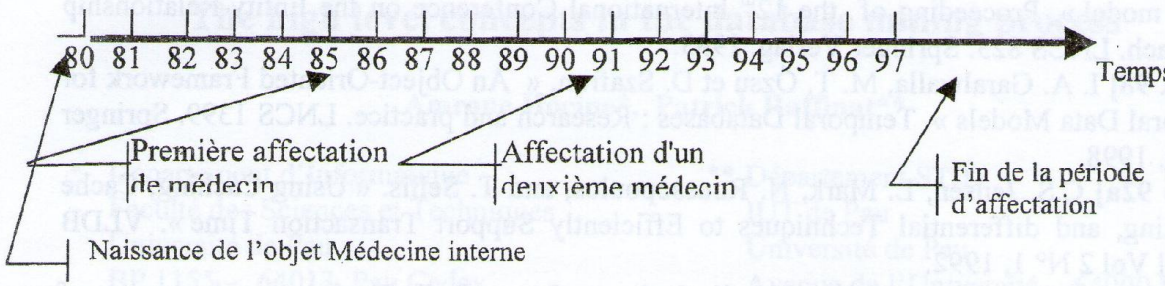


Dans le cas de l'association entre l'objet Médecin et l'objet Service, nous obtenons:



A travers ce schéma nous constatons :

a- le cycle de vie de l'objet Médecine interne est :



b- On distingue bien les périodes pendant lesquelles aucun médecin n'a été affecté au service médecine interne.
c- On donne la possibilité d'affecter un médecin a posteriori ou a priori
d- Les liens peuvent être définis au niveau des occurrences.

6. Conclusion

Dans ce qui précède, nous avons tenté de jeter les bases d'une démarche pour l'implémentation du temps dans les modèles à objets en exploitant la notion d'association. En donnant la possibilité au concepteur de modéliser l'évolution temporelle des objets et des liens entre objets, nous libérons le programmeur de la contrainte de la gestion des versions et nous nous donnons la possibilité d'un contrôle de cohérence de la base de données. Ainsi la prise en compte de l'adjonction du temps est intégrée dans le processus de modélisation et non pas au niveau application.
Notre démarche repose sur le principe de la nécessité de bien distinguer entre les différents niveaux de conception : conceptuel, logique et physique pour libérer le programmeur d'application de la gestion de mécanismes de chaînage des versions.
Les objectifs fixés se résument à :

1) Permettre la représentation explicite du temps dans la définition de l'objet en ciblant les parties de l'objet qui sont sujet à l'évolution.
2) Dégager les concepteurs des techniques d'archivage des informations qui évoluent en fonction du temps et assurer un degré de persistance des données suivant le choix du concepteur,
3) Offrir une stratégie d'expression des contraintes d'intégrité temporelles dans la définition de l'objet afin d'assurer la cohérence de la base de données.

L'automatisation de la gestion du temps nécessite le développement d'un SGBDOT, qui aura la particularité d'intégrer un gestionnaire d'objets temporels et un langage de manipulation enrichi par un ensemble de primitives temporelles spécialisées; c'est ce que nous nous fixons comme perspectives.

Références.

[BENZ 82] J. Ben-Zvi. « The Time Relational Model ». PhD Thesis, Computer Science Departement, UCLA (1982).
[CLIF 85] Clifford, J., and Tansel, A. « On an algebra for historical relationnel databases : Two views ». in ACM SIGMOD Conference (May 1985).
[CLIF 87] Clifford, J., and Crocker, A., « The Historical Relational Data Model (HRDM) and algebra based on lifespans ». in IEEE Data Engineering Conference (February 1987).
[CLIF 93] J. Clifford and A. Croker. « On an Algebra For the Historical Relational Data Model (HRDM) ». Dans Proceedings of the IEEE Data Engineering Conference, 1993.
[DAYA 92] Gene T.J. Wuu et Umeshwar Dayal. « A Uniform Model for Temporal Object-Oriented DataBases ». Dans Proceedings of the 8th IEEE Data Engineering Conference, 1992.
[DUBO 86] E. Dubois, j. Hagelstein and E. Lahou. « THE ERAE Model : a case study ». In Information Systems Design Methodologies : Improving the Practice (CRIS - 3). T. W. Olle, H. G. Sol and A. A. Verrijn - Stuart, eds. North - Holland, Amsterdam. 1986.
[GADI 88a] Gadia, S., and Yeung, C., « A generalized model for a temporal relational database », In ACM SIGMOD Conference (June 1988).

[GORA 93] I.A. Goralwalla et M.T. Ozsu, « Temporal extension to uniform behavioral object model ». Proceeding of the 12th International Conference on the Entity-Relationship Approach. LNCS 823. Springer Verlag, 1993.

[IQBA 98] I. A. Garalwalla, M. T. Özsu et D. Szafron, « An Object-Oriented Framework for Temporal Data Models ». Temporal Databases : Research and practice. LNCS 1399. Springer Verlag, 1998.

[JENS 92a] C.S. Jensen, L. Mark, N. Roussopoulos, and T. Sellis. « Using Caching, Cache Indexxing, and differential Techniques to Efficiently Support Transaction Time ». VLDB Journal Vol 2 N° 1, 1992.

[JENS 92b] C. S. Jensen, J. Clifford, S. K. Gadia, A. Segev, R. T. Snodgrass. « A glossary of temporal databases concepts » . ACM SIGMOD record, 21 (3), 1992.

[JENS 92c] C.S. Jensen, R.T. Snodgrass. « Temporal Specialization ». IEEE1992.

[JENS 94] C. Jensen, J. Clifford, R. Snodgrass. « A consensus glossary of temporal database concepts » . ACM SIGMOD record, 23 (1), Mars 1994.

[KAFE 92] W. Kafer et H. Schoning, « Realizing a Temporal Complex-Object Data Model ». Proceeding of the ACM SIGMOD Conference, California, juin 1992.

[KLOP 83] M. R. Klopproge. « TERM : an approach to include the time dimension in the ER model ». In 3rd Entity-Relationship Conference, Chen (october 1983).

[LORE 88] N. Lorentzos and R. Johnson. « Extending Relational Algebra to Manipulate Temporal Data ». Information Systems Vol 13 N° 3, 1988.

[MCKE 91] E. McKenzie. « An Algebraic Language for Query and Update of Temporal Databases ». PhD thesis, computer Science Departement, University of North Carolina 1988.

[NAVA 89] S. B. Navathe and R. Ahmed. « A Temporal Relational Model and a query Language ». Information Sciences N° 49, 1989.

[NAVA 90] Navath, S., and Ahmed, R., « A temporal data model and query language », Information Sciences (1990).

[ROSE 91] E. Rose, and A. Segev, « TOODM, a Temporal Object-Oriented Data Model with temporal contraints. ». Proceedings of the Tenth International Conference on the Entity Relationship Approach. 1991

[SARD 90] N. Sarda. « Extension to SQL for Historical Databases ». IEEE Transaction on Knowledge and Data Engineering Vol 2 N° 2, 1990.

[SEGE 87] Segev, A., and Shoshani, A., « Logical modeling of temporal data », In ACM SIGMOD Conference (June 1987).

[SNOD 86] R. T. Snodgras and I. Ahn. « Temporal Databases. » . IEEE Computer, Vol 19 No 9. 1986.

[SNOD 87] Snodgras, R., « The temporal query language TQUEL », ACM Transaction On Database Systems (TODS) Vol 12, N° 2 (June 1987).

[SNOD 93] R. Snodgrass. « Temporal object-oriented databases ». Rapport de recherche, Departement of Computer Science,University of Arizona, Mars 1993.S

[SNOD 94] R. Snodgrass et Al. « TSQL2 Language specification ». ACM SIGMOD Record, 23 (1), Mars 1994.

[SNOD 94a] R.T. Snodgrass, I. Ahn, G. Ariav, et al. « A TSQL2 Tutorial ». SIGMOD RECORD, Vol23, No 3, Septembre 1994.

[SNOD 95] Gultekin et R. Snodgrass. « Temporal and real-time databases : A Survey ». Dans Proceedings of the international conference on Data engineering.IEEE. 1995.

[SNOD 95a] R.T. Snodgrass, « Temporal Object-oriented databases : a critical comparison. » In Kim W. Ed, Modern database systems. The object model, interoperability and beyond, chaiptre 19. Addison Wesley, 1995.

[SUCH 91] S.Y.W. Su et H.H.M. Chen, « A Temporal Knowledge Representation Model OSAM*/T and its Query Language OQL/T ». Proceeding of the 17th VLDB international Conference, Barcelone 1991.

[TANS 86] A. U. Tansel. « Adding Time Dimension to Relational Model and Extending Algebra ». Information Systems Vol 11 N° 4, 1986.

[TANS 93] A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A.Segev, R. Snodgars et al « Temporal databases : theory design and implementation ». Benjamin/Cummings Publishing Company, Inc. 1993.