

100. Modélisation des données

Introduction

1 Niveaux d'abstraction retenus

La modélisation des données consiste à représenter les données utiles et nécessaires au SII. Le modèle ou la représentation des données n'est pas unique ; il y a autant de représentations ou de modèles qu'il y a de besoins à satisfaire à l'image des cartes routières qui donnent une indication sur les routes reliant les villes et les plans de ville qui représentent les quartiers et rue des villes. Les cartes et plans de ville se différencient par une indication d'échelle ; les modèles de données par un niveau d'abstraction.

Pour ce cours, nous retiendrons 3 niveaux d'abstraction inspirés des méthodes classiques telles que Merise ; ces 3 niveaux sont couramment utilisés par la communauté informatique.

- Le niveau conceptuel ou modèle conceptuel de données (MCD) définit la structure de données sur laquelle repose le SII sans contrainte économique, technique et d'organisation.
- Le niveau logique ou modèle logique de données (MLD) prépare un plan de mémorisation des données par le SII pour une classe de solution technologique indépendamment de la mise en œuvre par un produit spécifique.
Le modèle logique sera enrichi de caractéristiques qui lui sont propres et qu'il n'est pas judicieux de représenter au niveau conceptuel comme par exemple des champs d'audit qui permettent de savoir qui a manipulé quelle information.
- Le niveau physique ou modèle physique de données (MPD) réalise le plan de mémorisation des données en fournissant le code informatique de création des éléments techniques constitutifs de la structure de mémorisation des données du SII.
Le modèle physique sera enrichi de caractéristiques qui lui sont propres et qu'il n'est pas judicieux de représenter au niveau logique comme par exemple des privilèges utilisateurs.

Remarque : Le modèle physique de données (MPD) ne représente pas encore la réalité de la structure de stockage comme le laisse sous-entendre le terme usuel de niveau physique ; il s'agit bien d'un modèle en l'occurrence sous forme de code. La réalité ou le niveau physique ne sera effectivement atteint que suite à l'exécution du code.

2 Modèle conceptuel de données macroscopique (MCD)

A ce jour, les deux approches courantes de développement de logiciels de gestion préconisent de modéliser les éléments de données mais, chacune avec un modèle spécifique :

- l'approche classique s'appuie sur un modèle usuellement présenté sous le nom de « Modèle entités-associations »
- l'approche orientée objets s'appuie sur un modèle de classes nommé « Modèle du domaine ».

De plus, dans le cadre de l'approche classique, il existe une grande variété de formalismes; la communauté francophone est toujours très adepte du formalisme proposé par Merise alors que la communauté anglo-saxonne est plutôt adepte du formalisme de Chen ou Bachman.

Pour permettre un apprentissage de la modélisation conceptuelle des données indépendamment d'une approche ou d'une méthode spécifique, nous présenterons la modélisation conceptuelle des données dans une vision macroscopique ; c'est-à-dire que nous tenterons de mettre en évidence les concepts que nous retrouvons pratiquement dans les différentes propositions de modélisation conceptuelle des données.

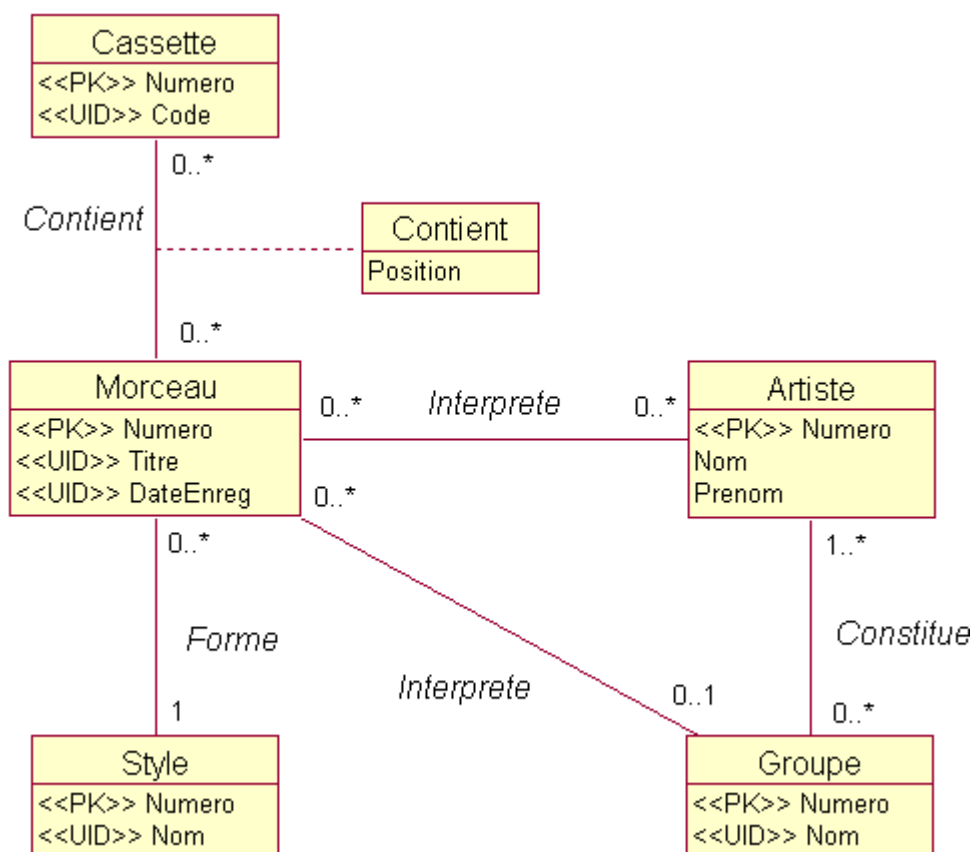


Figure 1 - MCD macroscopique de la gestion d'une cassetothèque réalisée avec Rational Rose

3 Modèle logique de données macroscopique (MLD)

Au fil du temps, plusieurs modèles logiques de données ont été proposés ; la montée en puissance des processeurs a permis, depuis les années 1990, la généralisation du modèle relationnel ; nous nous appuyons sur le modèle relationnel pour réaliser nos modèles logiques de données.

Le concept de modèle relationnel a été présenté par E. F. Codd en 1970 et fait l'objet, encore actuellement, de nombreux travaux de recherche.

[MERISE-3] En première approximation une relation peut être définie comme un **tableau de donnée**. Chacun de ces tableaux aura un nom et sera caractérisé par des **attributs**.

Le méta modèle s'inspire de la notion mathématique de relation, il est basé sur deux aspects fondamentaux:

- Aspect statique - Une démarche de conception permettant de définir une collection de relations aboutissant à la création d'un « Modèle logique de données relationnel ».
- Aspect dynamique - Une algèbre permettant de manipuler des tables ou relations.

Tout comme pour la modélisation conceptuelle, nous présenterons la modélisation logique dans une vision macroscopique du modèle relationnel ; c'est-à-dire que nous tenterons de mettre en évidence les concepts que nous retrouvons pratiquement dans les différentes implantations de modélisation logique des données.

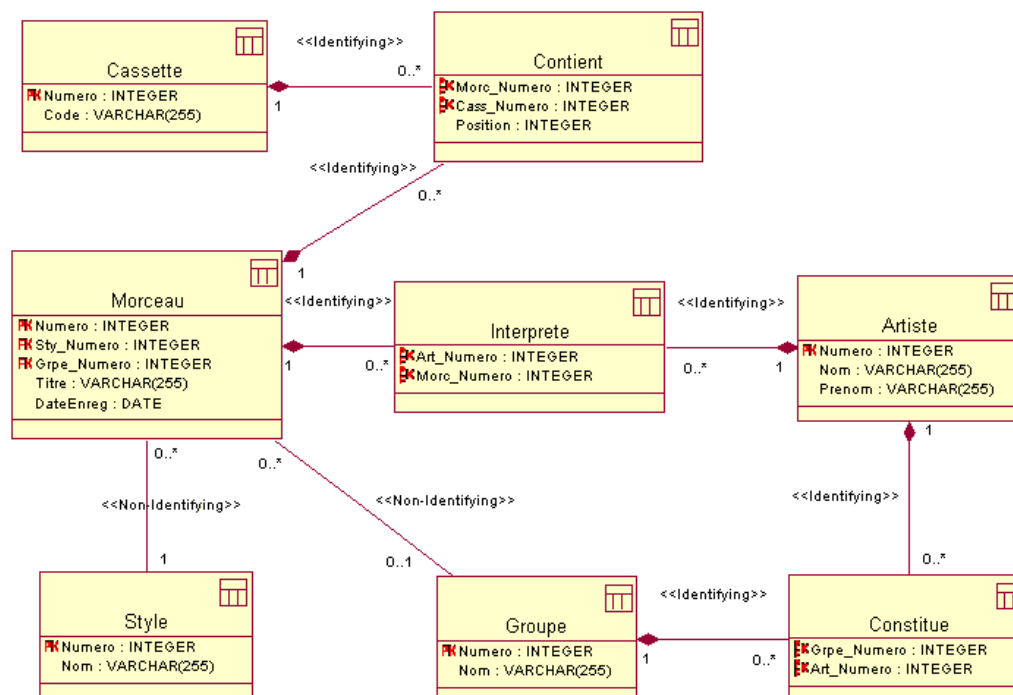


Figure 2 - MLD obtenu avec Rational Rose par la transformation automatique du MCD de la Figure 1

4 Modèle physique de données retenu (MPD)

A partir de modèles logiques de données réalisés selon le concept de modèle relationnel, il est naturel de choisir le langage normalisé SQL pour écrire les instructions qui permettront de créer physiquement le schéma de base de données relationnelle.

[NANCI-92] *Le langage de description et de manipulation de données adopté par la plupart des SGBD relationnels commercialisés est le langage SQL. Le langage SQL fut à l'origine le langage de consultation défini pour System R, SGBD prototype de recherche d'IBM. C'est l'acronyme de Structured Query Language. Ce langage a été normalisé dès 1986 (SQL1) puis en 1990 (SQL2 et SQL3).*

```
CREATE TABLE Artiste (  
    Numero INTEGER NOT NULL,  
    Nom VARCHAR ( 255 ) NOT NULL,  
    Prenom VARCHAR ( 255 ) NOT NULL,  
    CONSTRAINT PK_Artiste PRIMARY KEY (Numero)  
);  
CREATE TABLE Constitue (  
    Grpe_Numero INTEGER NOT NULL,  
    Art_Numero INTEGER NOT NULL,  
    CONSTRAINT PK_Constitue PRIMARY KEY (Grpe_Numero, Art_Numero)  
);  
CREATE TABLE Interprete (  
    Art_Numero INTEGER NOT NULL,  
    Morc_Numero INTEGER NOT NULL,  
    CONSTRAINT PK_Interprete PRIMARY KEY (Art_Numero, Morc_Numero)  
);  
CREATE TABLE Style (  
    Numero INTEGER NOT NULL,  
    Nom VARCHAR ( 255 ) NOT NULL,  
    CONSTRAINT PK_Style PRIMARY KEY (Numero)  
);  
CREATE TABLE Groupe (  
    Numero INTEGER NOT NULL,  
    Nom VARCHAR ( 255 ) NOT NULL,  
    CONSTRAINT PK_Groupe PRIMARY KEY (Numero)  
);  
CREATE TABLE Cassette (  
    Numero INTEGER NOT NULL,  
    Code VARCHAR ( 255 ) NOT NULL,  
    CONSTRAINT PK_Cassette PRIMARY KEY (Numero),  
    CONSTRAINT UK_CassetteCode UNIQUE (Code)  
);  
CREATE TABLE Morceau (  
    Numero INTEGER NOT NULL,  
    Sty_Numero INTEGER NOT NULL,  
    Grpe_Numero INTEGER,  
    Titre VARCHAR ( 255 ) NOT NULL,  
    DateEnreg DATE NOT NULL,  
    CONSTRAINT PK_Morceau PRIMARY KEY (Numero)  
);  
CREATE TABLE Contient (  
    Morc_Numero INTEGER NOT NULL,  
    Cass_Numero INTEGER NOT NULL,  
    Position INTEGER NOT NULL,  
    CONSTRAINT PK_Contient PRIMARY KEY (Morc_Numero, Cass_Numero)  
);  
CREATE INDEX FKI_ConstGrpe ON Constitue (Grpe_Numero);  
CREATE INDEX FKI_ConstArt ON Constitue (Art_Numero);  
CREATE INDEX FKI_IntArt ON Interprete (Art_Numero);  
CREATE INDEX FKI_IntMorc ON Interprete (Morc_Numero);  
CREATE INDEX FKI_MorcGrpe ON Morceau (Grpe_Numero);  
CREATE INDEX FKI_MorcSty ON Morceau (Sty_Numero);
```

```

CREATE INDEX FK_ContMorc ON Contient (Morc_Numero);
CREATE INDEX FK_ContCass ON Contient (Cass_Numero);
ALTER TABLE Constitue ADD CONSTRAINT FK_ConstGrpe FOREIGN KEY (Grpe_Numero) REFERENCES
Groupe (Numero) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE Constitue ADD CONSTRAINT FK_ConstArt FOREIGN KEY (Art_Numero) REFERENCES Artiste
(Numero) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE Interprete ADD CONSTRAINT FK_IntArt FOREIGN KEY (Art_Numero) REFERENCES Artiste
(Numero) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE Interprete ADD CONSTRAINT FK_IntMorc FOREIGN KEY (Morc_Numero) REFERENCES
Morceau (Numero) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE Morceau ADD CONSTRAINT FK_MorcSty FOREIGN KEY (Sty_Numero) REFERENCES Style
(Numero) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE Morceau ADD CONSTRAINT FK_MorcGrpe FOREIGN KEY (Grpe_Numero) REFERENCES
Groupe (Numero) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE Contient ADD CONSTRAINT FK_ContMorc FOREIGN KEY (Morc_Numero) REFERENCES
Morceau (Numero) ON DELETE NO ACTION ON UPDATE NO ACTION;
ALTER TABLE Contient ADD CONSTRAINT FK_ContCass FOREIGN KEY (Cass_Numero) REFERENCES
Cassette (Numero) ON DELETE NO ACTION ON UPDATE NO ACTION;

```

Figure 3 - MPD obtenu avec Rational Rose par la transformation automatique du MLD de la Figure 2

5 Transition entre niveaux d'abstraction

Les plans de ville ou les cartes de géographie quelle que soit leur échelle représentent toujours une portion de territoire qui est décrite avec plus ou moins de détail. Il est possible de créer un plan de ville à partir d'une carte géographique ; dans ce cas, la carte de géographie fournit les informations essentielles telles que les axes routiers principaux ou les cours d'eau, il faudra ensuite ajouter au plan de ville les détails tels que les rues de quartiers.

La démarche peut aussi être conduite dans l'autre sens ; à partir d'un plan de ville, il est possible de créer une portion de carte géographique qui synthétise certains traits principaux de la ville comme les cours d'eau et les axes routiers qui vont relier cette ville à d'autres « éléments » de la carte géographique.

La double démarche que nous venons de décrire se pratique naturellement aussi en modélisation des données.

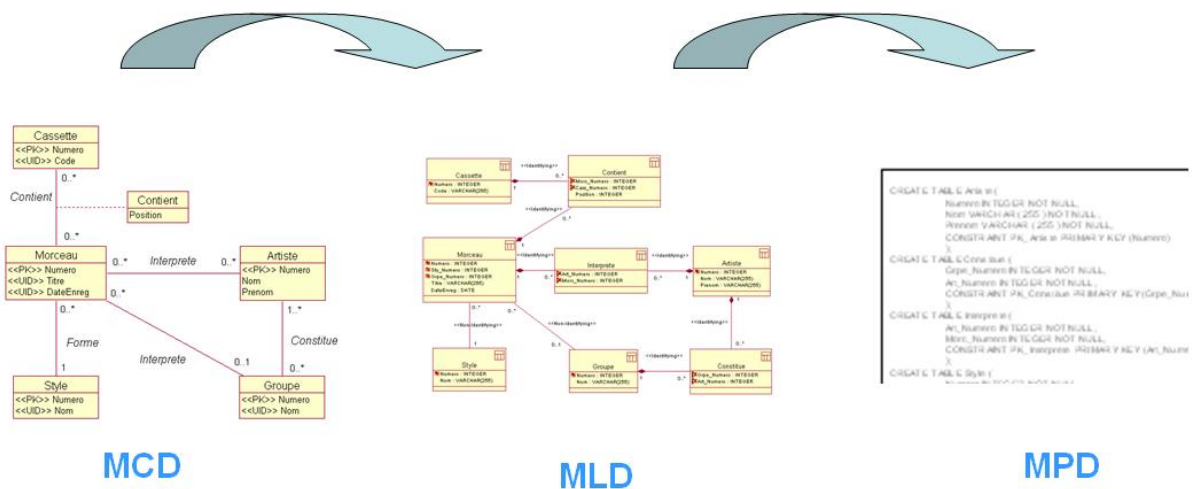


Figure 4 – Schéma des transitions entre niveaux d'abstraction dans le cadre d'une approche top/down

Dans une vision top/down, où nous partons du plus abstrait pour aller vers le plus concret, nous pouvons créer les MLD en appliquant aux MCD un algorithme éprouvé basé sur 3 règles de transformation. Ensuite, le passage du MLD au MPD consiste simplement à transcrire dans le langage SQL¹ le contenu du MLD.

Comme ces transformations obéissent toutes deux à des règles formelles, la plupart des ateliers de génie logiciel actuels effectuent automatiquement ces changements de niveaux de modélisation à la perfection².

Dans une vision d'ingénierie inverse, il est aussi possible de créer les modèles d'abstraction élevée à partir de modèles d'abstraction moins élevée ou de structures de données réelles; dans ce cas, tout comme pour les cartes géographiques, il y a lieu de choisir ce qui est pertinent de faire figurer dans le nouveau modèle d'abstraction plus élevée.

6 Qualité

De par l'emphase des données sur les traitements prônée par les technologies de l'information (TI) et très souvent appliquée par les différentes méthodes de l'approche classique ou de l'orienté objet, il est évident que la qualité du logiciel sera intimement lié à la qualité de la structure de données du SII ; si la structure de données est incohérente, il n'est pas possible de disposer de traitements qui rendent des résultats cohérents.

Qui dit : « modélisation conceptuelle des données de qualité » dit que cette activité de modélisation doit satisfaire les besoins d'un client. Quel est donc ce client et quels sont ces besoins?

Toujours en nous appuyant sur l'emphase des données sur les traitements, nous pouvons répondre que le client est la personne en charge de la modélisation conceptuelle des traitements et ses besoins sont de disposer d'une structure de données susceptible de supporter l'ensemble des traitements qu'il doit spécifier³.

En conséquence, pour qu'un logiciel de gestion satisfasse aux critères d'évaluation de la qualité, par exemple ISO 9126, il est impératif qu'en amont la structure de données satisfasse aux dits critères.

Dans le tableau ci-après, nous proposons quelques règles de modélisation des données pour que la structure de mémorisation des données participe à la satisfaction des besoins exprimés vis-à-vis du SII.

¹ Le langage SQL est ensuite compréhensible par les SGBD - Systèmes de gestion de bases de données.

² Demeure réservé, les changements applicables à des modèles ou schémas de base de données déjà existants ; dans ces cas-là, les règles de transformation sont plus complexes et tous les ateliers de génie logiciels ne savent pas faire une consolidation correcte tout en préservant l'existant.

³ Naturellement, la préséance des données sur les traitements n'implique pas que la modélisation des données soit complètement terminée avant de modéliser les traitements ; en général, la démarche s'effectue par des allers-retours successifs entre modélisation des données et modélisation des traitements.

Critères	Caractéristiques de la mémorisation des données
Capacité fonctionnelle	Toutes les données nécessaires aux traitements de l'information doivent faire partie de la structure ⁴ .
Fiabilité	Les données doivent être exemptes de redondance ⁵ . Les données issues du traitement d'autres données, des données dites calculées, doivent être identifiables et régénérables aisément ⁶ .
Facilité d'utilisation	Les données doivent être facilement repérables et identifiables à partir des différents modèles établis ⁷ .
Rendement	La mémorisation étant un élément statique, il n'y a, à priori, pas de critère de rendement, si ce n'est une erreur manifeste de structure.
Maintenabilité	Les données doivent être conçues en tenant compte des projets de l'organisme. Il ne faut pas se contenter de modéliser le présent, mais il y a lieu, dans la mesure du possible, d'anticiper les choix futurs de gestion.
Portabilité	En respectant les différents niveaux d'abstraction, la portabilité est assurée en conservant la stabilité par niveau. (Norme SQL pour les SGBD-R).

⁴ Il s'agit de s'assurer de la complétude du modèle

⁵ La vérification de la non-redondance de données fait l'objet de techniques spécifiques que nous traitons au [chapitre 170 – Normalisation & formes normales](#)

⁶ Nous traitons ce point en détail au chapitre 190 - Dénormalisation

⁷ Cette caractéristique ne peut être atteinte que par la mise en place et le strict respect de règles de nommage au sein d'une communauté, d'une entreprise ou d'un projet.