

# Modéliser des données, l'exemple du modèle entité-association

*Modéliser les données d'un système logiciel, c'est construire une structure de données contraintes, représentative du réel observé.*

Lorsqu'on analyse les informations véhiculées dans un système d'information de gestion, on observe certes le « réel », directement (analyse de documents ou d'une application existante, par exemple) ou indirectement (entrevues avec des utilisateurs, par exemple). Cette référence au réel ne doit pas être oubliée sous peine de construire des systèmes d'élucubration ! Il n'en demeure pas moins que l'analyse et la conception se fondent, à quelque niveau que ce soit, sur des représentations, autrement dit des abstractions :

- On ne retient pas tout ce qu'on peut observer : on retient ce qui formalise le point de vue de l'organisation.
- On ne recense pas tous les exemplaires passés ou présents, encore moins à venir ! On recherche des similitudes, on élabore des types.

« Un objet est pensé et non pas senti. »

Alain

Lorsqu'on conçoit un système logiciel, on fait l'hypothèse<sup>1</sup> que l'ensemble des données retenues est « structurable » c'est-à-dire que les éléments retenus entretiennent des rapports entre eux et que la nature de chaque rapport peut être déterminée puisqu'il exprime... quelque chose.

Dans tous les domaines de la connaissance, on procède ainsi. L'étude des méthodes de conception de systèmes logiciels en fournirait une nouvelle preuve :

- On chercherait à établir des typologies, c'est-à-dire à distinguer dans l'ensemble des méthodes certaines « catégories » : les méthodes cartésiennes, les méthodes systémiques, ...
- On donnerait des exemples : Merise est une méthode systémique.
- On montrerait aussi qu'une méthode est « composée », par exemple de quatre types d'éléments : des modèles, des langages, des démarches, des outils.

Catégorisation, instanciation, composition font partie des types d'abstraction les plus usités pour structurer nos connaissances. Pour apprendre à modéliser, il faut comprendre ce qu'apporte chacun de ces types d'abstraction, quelles règles structurelles les gouvernent, donc sur quels implicites ils reposent, et quelles précisions ils permettent d'explicitier. Pour apprendre à modéliser, il faut en d'autres termes apprendre à structurer et à exprimer des contraintes.

---

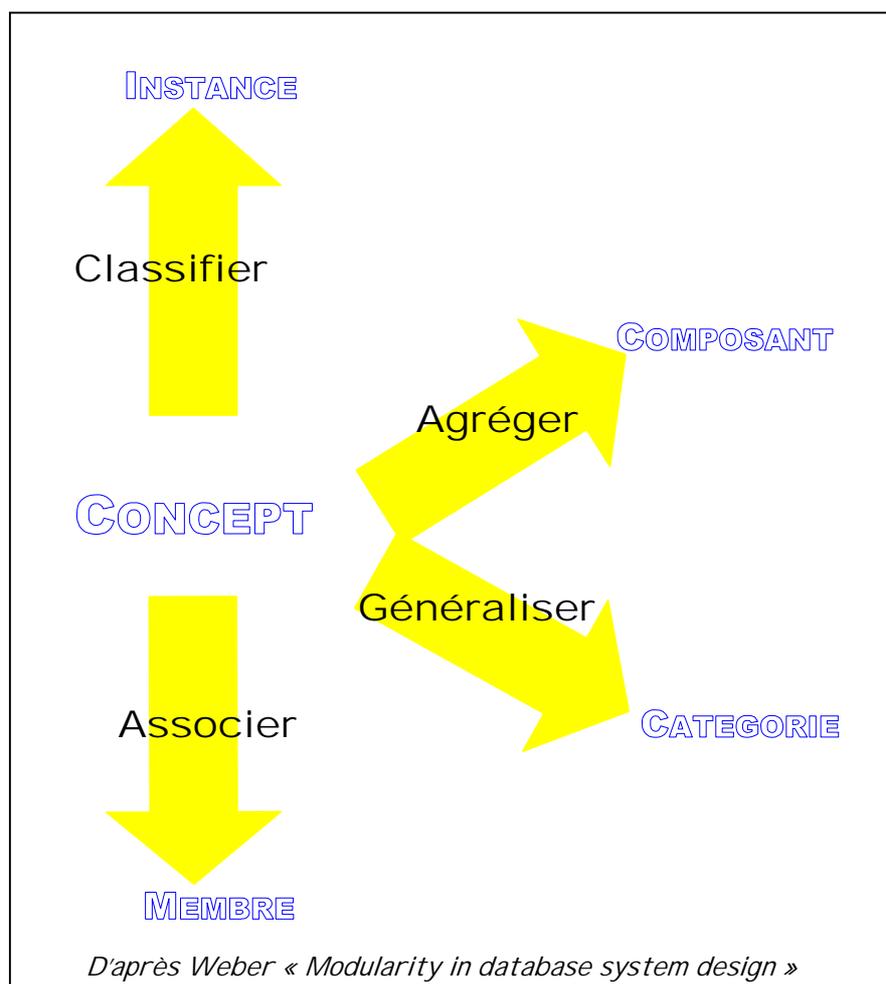
<sup>1</sup> Hypothèse justifiée (comme dans toute activité humaine « industrielle ») par le fait que les systèmes qu'elle permet de construire répondent aux besoins énoncés.

## Concrètement, c'est quoi l'abstrait ?

### I - Structurer

L'abstraction est une capacité typique de l'être humain : sa maîtrise de structures et de procédures complexes ne serait pas aussi élevée s'il n'avait pas su doter sa mémoire (au sens dynamique du terme) de facultés typiques d'abstraction, qu'il utilise quotidiennement pour accumuler et restituer de façon intelligible ses connaissances.

Qu'il s'agisse de sciences exactes, de philosophie, de linguistique ou d'informatique, on retrouve des conceptions similaires. De façon schématique, l'être humain classe et instancie, compose (ou agrège) et décompose, généralise et spécialise, groupe (ou associe) et individualise.



Attention, pour lire les lignes qui suivent, vous devrez parfois faire... abstraction de vos conceptions antérieures : ainsi, « agréger » et « composer » sont des termes présentés ici comme des synonymes, alors qu'ils se distinguent dans UML, où ils représentent une catégorisation du verbe « associer », etc. Avant de vous en prendre aux étiquettes choisies, nous vous invitons à examiner le contenu du colis.

Le langage « naturel » utilise les types d'abstraction comme Jourdain la prose. Quand le discours se porte sur les abstractions elles-mêmes, il est habituel d'en fournir des représentations à l'aide d'un langage particulier, utilisant des littéraux ou des symboles graphiques.

### La problématique

#### Exemple n° 1 : « Jean est un homme. »

Considérez cette affirmation triviale.

Jean n'est pour l'instant qu'une abstraction, pour chacun d'entre nous. Mais, intuitivement, vous comprenez qu'il s'agit de parler d'un particulier en général (qui peut très bien, au demeurant, être général en particulier ☺). Soit il s'agit de caractériser Jean en faisant appel à un concept partagé entre nous (la notion d'homme), soit il s'agit d'expliciter le concept d'homme en donnant un exemple connu de nous tous. Quel que soit le sens de la pensée, il y a un type de rapport singulier entre « Jean » et « homme » : ce type de rapport sémantique peut s'appeler la classification.

L'affirmation considérée peut signifier :

- « Jean appartient à l'ensemble des hommes. »
- « Jean est un élément de l'ensemble des hommes. »

Cette notion primaire est importante. Si le type « homme » est défini, à l'aide de caractéristiques typiques de formes et de comportement (le nom, le prénom, l'adresse, le mois de naissance, la religion, la profession, le lieu de travail, le vêtement préféré, si marié : avec qui ?, etc.) et si Jean reste, au regard de cette définition, un homme, c'est qu'il possède une expression particulière des caractéristiques évoquées (Jean, Jean, rue Jean XXIII à Janville, janvier, janséniste, jean-foutre, Compagnie des Gens, jean, marié avec... j'en sais rien !). Si l'ensemble des hommes partage collectivement d'autres caractéristiques, Jean en profitera. De même, en établissant les droits et les devoirs du citoyen français (type), on s'assure (en principe !) que chaque citoyen (occurrence) possède les mêmes droits et les mêmes devoirs.

### Le concept

La notion de classification est évidemment partagée par de nombreux domaines de connaissances, sous des vocables qui peuvent différer.

	PHILOSOPHIE	BIOLOGIE	MATHÉMATIQUES	MÉTHODE MERISE	LANGAGE OBJET
HOMME	Concept	Espèce	Ensemble	Type	Classe
JEAN	Réalisation	Spécimen	Élément	Occurrence	Instance <sup>2</sup>

#### CLASSIFICATION

*Abstraction par laquelle on perçoit une classe d'objets comme un ensemble de réalisations (appelées instances) d'un type d'objet représentatif de la classe.*

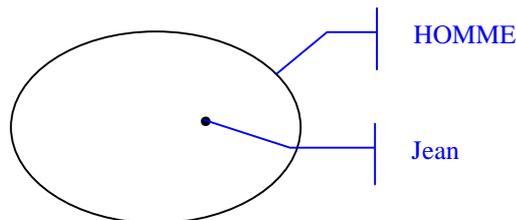
<sup>2</sup> Ou objet, bien que certains réservent ce terme au monde réel. De plus, dans son acception abstraite, Objet est également une classe et la classe est un objet ! On aurait pu également citer les moules utilisés par diverses industries et la... couture avec ses patrons et ses exemplaires.

En matière d'abstraction, l'instanciation est le phénomène de pensée inverse de la classification. Dans l'activité de modélisation, c'est le premier pas que franchit la pensée : par la suite, on ne se réfère plus guère aux choses sensibles mais aux abstractions que la classification a générées.

## Exemples de représentation

➤ Le langage mathématique en offre plusieurs :

- de façon littérale :  $Jean \in \text{HOMME}$
- « Choses » représentées graphiquement :



➤ Voici d'autres exemples :

- Entièrement littéraux :  $Jean \text{ is\_a } \text{HOMME}$  ou  $Jean \text{ est\_un } \text{HOMME}$
- Lien représenté graphiquement :  $Jean \xrightarrow{\dots\dots\dots} \text{HOMME}$

## 1.2 - La composition (ou agrégation)

### La problématique

#### Exemple n° 2 : « Un homme est formé d'un corps, d'une tête et de quatre membres. »

Considérez cette affirmation infantilement anatomique (genre Toto ☺).

Elle est bien d'un genre radicalement différent de la précédente. On y mentionne un concept considéré d'une part comme un tout (le concept d'homme), d'autre part comme le composé d'autres concepts plus simples (les concepts de corps, de tête et de membre). L'objet offre deux perspectives : d'abord celle des composants de nature diverse qu'il intègre, mais aussi celle d'un agrégat<sup>3</sup> perçu comme un objet particulier<sup>4</sup>.

Dans l'exemple, l'assemblage décrit, pour abstrait qu'il soit, renvoie à une réalité tangible. Dans les systèmes de gestion, la réalité, c'est l'information sur les « choses », des objets mais aussi des événements, des « choses » qui concernent une organisation particulière, poursuivant des objectifs particuliers, toutes choses qui fondent donc un point de vue particulier.

<sup>3</sup> Attention à l'emploi de ce mot dans d'autres disciplines ! En Économie, on parle d'agrégats à propos de la Comptabilité Nationale : ces agrégats rassemblent des données monétarisées (donc de même nature) alors qu'ici les éléments rassemblés peuvent être de nature différente.

<sup>4</sup> « Savoir, ou plutôt pouvoir affirmer le *quod*, sans aucun *quid* (l'existence d'une chose sans aucune détermination, aucun attribut), serait-ce *connaître* ? » (J. Lachelier, Vocabulaire technique et critique de la philosophie, tome I, note p. 171).

« Qu'est-ce qu'un homme pour l'état civil ? Pour une société de vente par correspondance ? Pour une agence matrimoniale ? Pour un négrier ? Pour un bourreau ? »

*des points de vue très différents*

## Le concept

Les notions de composé et de composant sont importantes en matière de modélisation. Elles permettent de préciser ce qui a été dit à propos de la définition d'une « classe » à l'aide de caractéristiques de forme. Ainsi, si la classe « homme » est définie à partir des caractéristiques appelées « nom » et « prénom », toute instance d'homme possédera une « instance de nom » et une « instance de prénom ». En modélisation, c'est en associant la classification et la composition que le concept de valeur prend tout son sens.

On utilise de nombreux vocables pour exprimer ce type de rapport entre diverses connaissances<sup>5</sup> :

<b>HOMME</b>	Composé	Molécule	Entité	Classe	Tout
<b>CORPS, TÊTE, MEMBRES</b>	Composants	Atomes	Propriétés	Attributs <sup>6</sup>	Parties

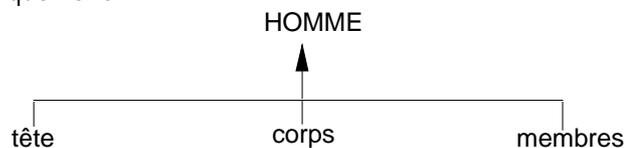
### COMPOSITION

*Abstraction par laquelle un lien entre objets de types divers est perçu comme un type d'objet à part entière.*

L'inverse de la composition (ou agrégation) est la décomposition (ou description).

## Exemples de représentation

- Entièrement littéraux : HOMME(tête, corps, membres)  
tête *part\_off* HOMME ou tête *partie\_de* HOMME
- Lien représenté graphiquement :



Remarques :

- Le lecteur a l'intuition que, dans les représentations littérales, sont utilisées des conventions d'écriture. Par exemple : le nom d'un composé est écrit en lettres majuscules, le nom d'un composant en lettres minuscules.
- La représentation de la composition à l'aide d'un « peigne » (ou « fourchette ») est fréquente dans diverses disciplines. Ce n'est pas forcément la meilleure puisque ce mode de représentation est utilisé pour exprimer d'autres types de liens (cf. suivant).

<sup>5</sup> Ici, les vocables traversent souvent les frontières des domaines de connaissances concernés.

<sup>6</sup> Ou encore rubriques, champs, zones...

## I.3 - La généralisation

### La problématique

#### Exemple n° 3 : « L'homme est un mammifère. »

Considérez cette affirmation tout aussi banale que l'exemple n° 1.

L'emploi du même groupe verbal (« est un ») dans les deux exemples peut troubler. Peut-on dire, dans l'acception mathématique des termes employés, que l'homme est un élément de l'ensemble des mammifères ? Que désigne-t-on par l'homme ? Un homme en particulier (Jean, par exemple) : dans ce cas, il est bien un élément de l'ensemble des mammifères. Mais il semble bien qu'ici on ait voulu désigner l'homme en général. L'affirmation veut probablement dire :

- « L'ensemble des hommes est inclus dans l'ensemble des mammifères. »
- « L'ensemble des hommes est un sous-ensemble de l'ensemble des mammifères. »

Ce faisant, on s'est livré à un type d'abstraction appelé généralisation.

Cette capacité est fondamentale dans l'organisation des connaissances, à tel point qu'elle est à l'origine de la structuration des thesaurus qui permettent la recherche documentaire. On dira : « Homme est un spécifique de Mammifère qui est son générique. »

- Si le type « mammifère » est défini à l'aide de caractéristiques de forme et de comportement, le type « homme » possédera les mêmes types de caractéristiques : il en héritera.
- Des exemples n° 1 et n° 3, vous pouvez déduire (brillamment ☺) que Jean est un mammifère : en associant la classification et la généralisation, on peut faire de « l'inférence » c'est-à-dire faire naître une nouvelle connaissance.

### Le concept

La généralisation est une notion partagée par de nombreux domaines de connaissances, sous des vocables qui peuvent différer.

	PHILO-SOPHIE	BIOLOGIE <sup>7</sup>	MATHÉ-MATIQUES	MÉTHODE MERISE	LANGAGE OBJET
MAMMI FÈRE	Concept	Genre	Ensemble	Type	Classe
HOMME	Catégorie	Espèce	Sous-ensemble	Sous-type	Sous-classe

#### GÉNÉRALISATION

*Abstraction par laquelle un type est défini à partir des caractéristiques communes à plusieurs autres types.*

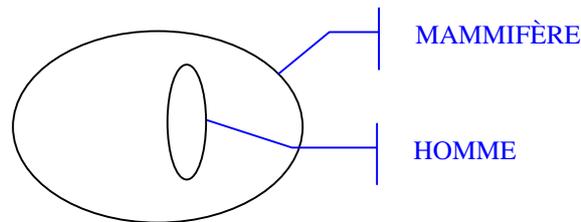
<sup>7</sup> Nous avons choisi ces termes parce qu'ils prolongent ceux qui apparaissent dans le premier tableau. Ils ne sont pas vraiment adaptés puisque les mammifères forment en fait une classe. La hiérarchie des catégories est « naturellement » la plus évoluée en sciences... naturelles : classe (mammifères), [sous-classe], ordre (primates), [sous-ordre, groupe (hominies), superfamille], famille (hominidés), genre (*homo*), espèce (*homo sapiens*), [sous-espèce, race, ...].

En première approche, la spécialisation est l'inverse de la généralisation<sup>8</sup>.

## Exemples de représentation

➤ Le langage mathématique en offre plusieurs :

- de façon littérale :  $\text{HOMME} \subseteq \text{MAMMI FÈRE}$
- Ensembles représentés graphiquement :

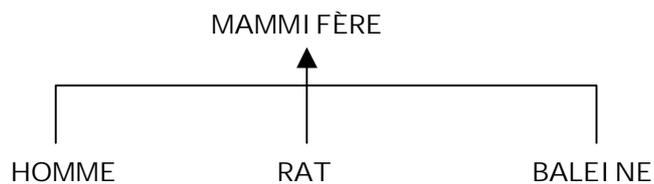


➤ Voici d'autres exemples :

- Entièrement littéraux :  
 $\text{HOMME } \textit{kind\_of} \text{MAMMI FÈRE}$   
 $\text{HOMME } \textit{sorte\_de} \text{MAMMI FÈRE}$   
 $\text{HOMME} : \text{MAMMI FÈRE}$
- Lien représenté graphiquement :  $\text{HOMME} \longrightarrow \text{MAMMI FÈRE}$

Remarque :

- On peut avoir à représenter plusieurs sous-ensembles d'un même ensemble. Dans ce cas, le « peigne » est quelquefois utilisé (UML, par exemple, mais avec une pointe creuse).



Le choix des représentations est bien sûr guidé par l'absence d'ambiguïté : le « peigne » ne peut à la fois représenter la composition et la généralisation qui sont deux concepts bien différents.

---

<sup>8</sup> La spécialisation, à l'instar de la composition, exprime un point de vue. Fonder une typologie se fait en adoptant un certain regard : ainsi, l'homme et le castor peuvent se ressembler en ce sens que tous deux sont des mammifères ou bien qu'ils modèlent leur environnement plutôt que de s'y adapter ou encore qu'ils ne travaillent ni avec leurs mains ni avec leurs pieds !

## I.4 - La construction d'ensemble

### La problématique

Exemple n° 4 : « Jumbo est un éléphant. »

Exemple n° 4 bis : « L'éléphant est une espèce en voie de disparition. »

Ces deux énoncés nous rappellent les exemples n° 1 et n° 3 :

Exemple n° 1 : « Jean est un homme. »

Exemple n° 2 : « L'homme est un mammifère. »

De ces deux énoncés, vous aviez pu déduire : « Jean est un mammifère. ». Des énoncés 4 et 4 bis, que déduire ? Que : « Jumbo est une espèce en voie de disparition » ?

En fait, l'emploi, dans tous ces énoncés, de l'expression « est un » cache une différence fondamentale.

- Dans le premier cas, l'inférence est possible car la première affirmation relate un lien d'appartenance ( $\text{Jean} \in \text{HOMME}$ ) et la deuxième un lien d'inclusion ( $\text{HOMME} \subseteq \text{MAMMI FÈRE}$ ).

Jean .....→ HOMME → MAMMI FÈRE

- Dans le deuxième cas, nous n'avons que des liens d'appartenance, ce qui ne nous permet aucune inférence (non tautologique).

Jumbo .....→ ÉLÉPHANT .....→ ESPÈCES MENACÉES

ESPÈCES MENACÉES est un ensemble d'ensembles. ÉLÉPHANT, qui est un ensemble (une classe), est un élément (un objet, une instance) de ce premier ensemble.

Dans de nombreux domaines de connaissances, on a besoin de cette faculté de considérer une classe d'objets comme un objet d'une autre classe, de considérer l'ensemble comme un individu.

Exemple : En droit, la notion de personnalité « morale » est proche de cette notion. Une société est composée d'associés : en ce sens, c'est un ensemble de « personnalités physiques ». Mais la société possède également une existence juridique propre.

### Le concept

Il n'est pas aisé de donner une étiquette à ce type d'abstraction. Dans certaines disciplines, on le nomme « association », par analogie avec l'association qui, en France, relève de la loi 1901 et qui s'intègre dans la problématique de personnalité morale évoquée ci-dessus. Compte tenu de l'ambiguïté de ce terme, nous lui préférons l'expression de « construction d'ensemble ».

#### CONSTRUCTION D'ENSEMBLE

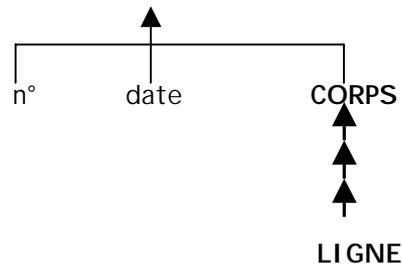
*Abstraction par laquelle un ensemble d'objets est perçu comme un objet.*

Cette faculté est peu utilisée dans les modèles de données intégrés aux méthodes de conception de systèmes d'information. Elle induirait pourtant des descriptions (décompositions) souvent plus naturelles : par exemple, on peut décrire une commande par un numéro, une date, etc. (éléments caractéristiques de son en-tête) et son corps qui est un ensemble de lignes de commande. Cette possibilité est offerte par UML qui permet de considérer un objet comme une collection : un ensemble (sans doublon), un sac (avec doublons) ou une séquence (ordonnée).

## Exemples de représentation

- Entièrement littéraux :  
 COMMANDE(n°, date, CORPS)  
 CORPS = {LIGNE}

- Lien représenté graphiquement : COMMANDE



## I.5 - Et l'association ?

### La problématique

**Exemple n° 5 : « Jean, qui est un homme, dresse Jumbo, qui est un éléphant. »**

Dans cet énoncé, vous considérez une instance d'homme et une instance d'éléphant. Entre les deux, un rapport existe, représenté par l'emploi du verbe « dresser ». Au niveau des classes d'objets, comment représenter un point de vue qui s'intéresserait à « qui dresse qui ? » ?

### Le concept

Dans de nombreux modèles, on part du principe que ce type de lien est à représenter de façon particulière, pour distinguer :

- d'une part les classes d'objet qui ont une existence propre, qui sont autonomes : elles sont souvent désignées à l'aide de substantifs (exemples : HOMME, ÉLÉPHANT) ;
- d'autre part les rapports que les instances de ces classes établissent entre elles<sup>9</sup> et qui ont un sens dans le point de vue à modéliser : ils sont souvent exprimés à l'aide de verbes. Le concept utilisé s'appelle le plus souvent association<sup>10</sup> (exemple : dresser).

Qu'il s'agisse d'un type d'abstraction indispensable ne fait pas l'unanimité ! Ainsi, le verbe « dresser » peut être substantivé en « dressage » et le concept de « dressage » décrit par un

<sup>9</sup> Dans certains modèles à objets, on parle d'ailleurs de « connexion d'instances ».

<sup>10</sup> Le terme « relation » utilisé à l'origine en France pour traduire le vocable américain *relationship* est tombé en désuétude du fait de sa polysémie (la notion mathématique de « relation » est à l'origine des systèmes de gestion de bases de données relationnels très répandus aujourd'hui ; or, il s'agit d'une notion radicalement différente).

dresseur (qui est un homme) et un dressé (qui est un éléphant). Dressage est un composé et ses composants se définissent par rapport à d'autres concepts. Il existe cependant deux façons d'interpréter cette définition :

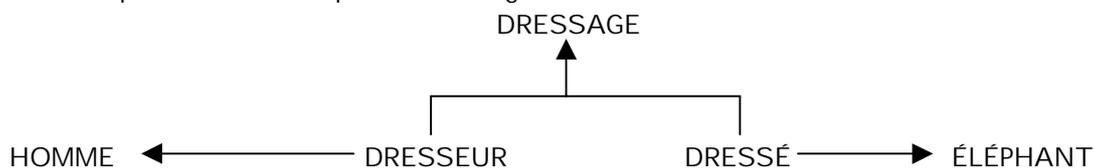
- Si l'on considère l'ensemble des occurrences de dressage, l'ensemble des dresseurs correspondants constituent un sous-ensemble des hommes. Un lien de généralisation s'établit entre les deux notions.
- Si l'on considère une occurrence de dressage, le dresseur correspondant est un élément de l'ensemble des hommes. Un lien de classification s'établit entre les deux notions. Dans les langages objets, on fait généralement cette distinction entre la définition d'une sous-classe et la définition d'un attribut dont le type fait référence à une classe.

## Exemples de représentation

- Reposant sur le concept d'association :



- Reposant sur la composition et la généralisation :



## 1.6 - Le modèle entité-association

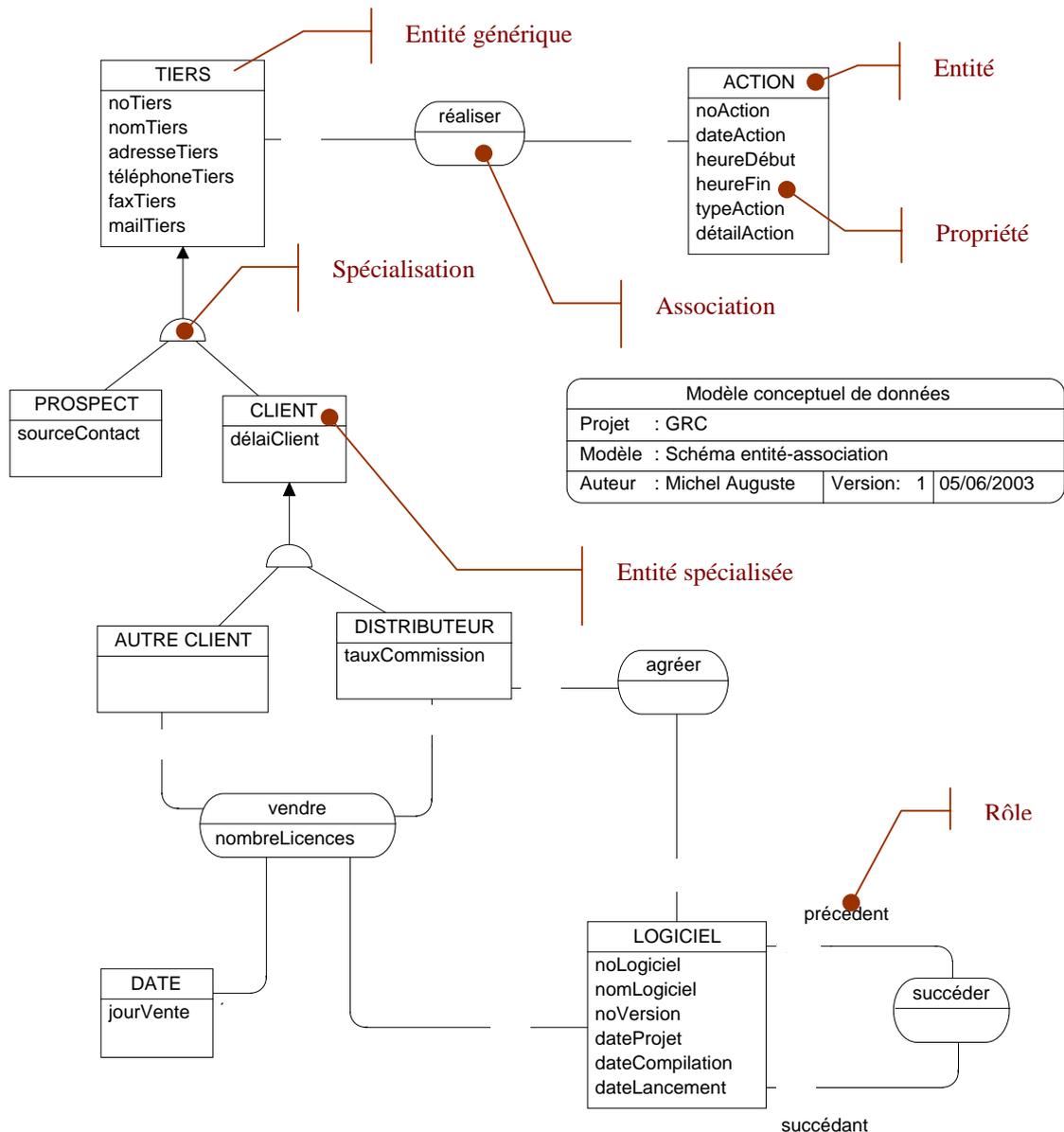
D'un certain point de vue, le modèle entité-association (MEA) né au milieu des années 1970 repose sur deux niveaux de composition :

- Les propriétés sont les briques de base.
- Les entités sont composées de propriétés.
- Les associations transcrivent les rapports que les entités entretiennent entre elles. Comme il vient d'être dit, elles peuvent être interprétées en termes de composition : une association est alors composée d'entités et, éventuellement, de propriétés.

La capacité sémantique du modèle a été enrichie dès la fin des années 70 aux États-Unis : les extensions majeures concernent l'agrégation (envisagée ici comme un 3<sup>e</sup> niveau de composition, elle offre donc la possibilité de construire des « associations d'associations ») et la spécialisation<sup>11</sup>. Les modèles utilisés en France ont connu la même évolution plus tardivement, dans les années 90.

Le schéma entité-association présenté à la page suivante illustre certaines de ces notions. Il modélise partiellement un système de gestion de la relation client (GRC) en s'inspirant largement d'un travail effectué par Karine Cubizolles, étudiante à l'I UFM de Bourgogne (promotion 2003).

<sup>11</sup> Dans ce document, la spécialisation sera considérée comme l'inverse de la généralisation. En fait, dans la plupart des modèles utilisés en France, on distingue la spécialisation, qui permet au concepteur de prendre en compte les catégories mentionnées par l'utilisateur, et la généralisation, qui permet au concepteur de construire des entités abstraites à partir de concepts que le discours de l'utilisateur n'intègre pas.



## La sémantique représentée

- Le système GRC gère des tiers. Chaque tiers est décrit par un numéro, un nom, une adresse postale, un n° de téléphone, un n° de fax et une adresse électronique.
- Parmi ces tiers, on distingue notamment deux catégories : les prospects caractérisés par l'indication de la source de la mise en relation et les clients pour lesquels on a besoin de connaître le délai de paiement accordé.
- Parmi ces clients, on distingue les distributeurs auxquels on accorde un taux de commission particulier et les autres clients.
- Des actions sont réalisées auprès des tiers. Une action est caractérisée par son numéro, la date de réalisation, l'heure début et l'heure fin de l'action, le type d'action réalisée (technique, commerciale, financière, ...) et le détail de l'action sous forme de texte.
- Les logiciels commercialisés sont décrits par un numéro, un nom, un n° de version et des dates de début de projet, de compilation et de lancement.
- Un logiciel peut succéder à un autre logiciel.
- Les distributeurs sont agréés pour distribuer certains logiciels.
- Les distributeurs vendent des logiciels aux autres clients à certaines dates.

## Les composantes de la représentation graphique

- Les rectangles à cartouche représentent les entités (ou les sous-types d'entités). Le nom de l'entité est placé dans le cartouche : c'est généralement un substantif.
- Les ellipses à cartouche représentent des associations. Le nom de l'association est placé dans le cartouche : c'est généralement un verbe.
- Lorsqu'une même entité joue plusieurs rôles dans une association (par exemple, l'association « succéder »), les différents rôles se distinguent par leurs noms (par exemple, « succédant » et « précédent »). Mais on aurait pu généraliser l'emploi du nom de rôle pour toutes les associations : le trait qui relie une entité à une association, souvent appelée « patte », représente précisément un rôle.
- Chaque propriété est placée dans son entité ou bien dans son association.
- La demi-lune représente une spécialisation<sup>12</sup>. La flèche pointe vers l'entité générique.

Il faudrait compléter le schéma par un « dictionnaire de données », autrement dit une caractérisation précise des propriétés citées. Dans ce document, nous éludons volontairement cette question.

Dans un modèle donné, chaque désignation utilisée (nom de propriété, nom d'entité, nom d'association) est unique.

## Les représentations littérales

- Voici une représentation possible :

```
TIERS(noTiers, nomTiers, adresseTiers, téléphoneTiers, faxTiers, mailTiers)
ACTI ON(noAction, dateAction, heureDébut, heureFin, typeAction, détailAction)
Réaliser(TIERS, ACTI ON)
PROSPECT(sourceContact) : TIERS
CLI ENT(délaiClient) : TIERS
AUTRE CLI ENT : CLI ENT
DI STRI BUTEUR(tauxCommission) : CLI ENT
DATE(jourVente)
LOGI CI EL(noLogiciel, nomLogiciel, noVersion, dateProjet, dateCompilation, dateLancement)
Agréer(DI STRI BUTEUR, LOGI CI EL)
Vendre(DI STRI BUTEUR, AUTRE CLI ENT, DATE, LOGI CI EL, nombreLicences)
Succéder(LOGI CI ELsuccédant, LOGI CI ELprécédent)
```

- Les règles d'écriture utilisées.
  - Le nom d'une entité est écrit en lettres majuscules.
  - Seule, l'initiale du nom d'une association est une lettre majuscule.
  - Le nom d'une propriété est composé de morphèmes. Chaque morphème possède une initiale majuscule, sauf le premier.
  - Le lien de composition est représenté comme suit. Le composé est d'abord mentionné puis ses composants sont placés entre parenthèses, chaque composant étant séparé du précédent par une virgule.
  - Le nom de rôle est placé en indice et vient suffixer le nom de l'entité concernée.
  - Le lien de généralisation est représenté par le symbole « deux-points ».

---

<sup>12</sup> Il existe bien d'autres modes de représentation. Celui-ci a été choisi par les concepteurs du logiciel Power\*AMC utilisé pour construire les schémas.

- Les opérations élémentaires

Cette écriture se prête mieux aux calculs. En voici quelques exemples :

- L'ensemble des noms de tiers peut être obtenu par l'opération `TIERS[nomTiers]` qui se lit « Tiers projeté sur nomTiers ».
- L'ensemble des actions réalisées le 20-11-2003 peut être obtenu par l'opération `ACTION[dateAction = '20-11-2003']` qui se lit « Actions pour lesquelles dateAction égale 20-11-2003 ».
- L'ensemble des logiciels vendus au moins une fois peut être obtenu par l'opération : `Vendre[LOGICIEL]` qui se lit « vendre projeté sur LOGICIEL »<sup>13</sup>.
- On peut également utiliser des variables. Par exemple : `soit A ∈ ACTION`. A représente une occurrence de l'entité ACTION. On peut alors accéder à ses propriétés en utilisant la notation pointée. Par exemple : `A.dateAction`.

### Les capacités d'abstraction utilisées

- La composition est utilisée sur deux niveaux : l'entité et l'association.
  - L'entité TIERS, par exemple, est composée à partir des propriétés noTiers, nomTiers, adresseTiers, téléphoneTiers, faxTiers, mailTiers<sup>14</sup>.
  - L'association « vendre », par exemple, est composée à partir des entités DISTRIBUTEUR, AUTRE CLIENT, LOGICIEL, DATE et de la propriété nombreLicences.
- La spécialisation est également utilisée<sup>15</sup>. En conséquence, pour décrire complètement un distributeur, il faut prendre en compte le fait qu'il est un client, ce qui implique l'héritage de la propriété délaiClient, et qu'un client est un tiers, ce qui implique l'héritage de toutes les propriétés de cette entité.

Cette description est très incomplète. Par exemple, vous voudriez qu'on précise que noTiers identifie les tiers ou bien qu'une action n'est réalisée qu'après d'un seul tiers ou encore que les clients ne peuvent pas être des prospects. Et vous avez raison : les capacités d'abstraction mobilisées nous aident certes à établir les liens structurels qui existent entre les différents concepts mais cela ne suffit pas. De fait, chaque type de lien est caractérisé par des « décorations » typiques, qui permettent de préciser sous quelles contraintes le lien structurel s'établit dans la réalité. En cette matière, il existe des références mathématiques sur lesquelles nous prendrons appui après vous en avoir rafraîchi la mémoire.

---

<sup>13</sup> Nous faisons l'hypothèse que le résultat est un ensemble (ne comportant pas de doublons).

<sup>14</sup> Mettre en avant la composition n'est qu'une façon de voir l'activité de conception. Cela ne présume en rien de la primauté, sur le plan conceptuel, du composant sur le composé. Au contraire ! On aurait pu dire, peut-être plus justement, que la notion de TIERS est fondée d'abord par les besoins de gestion et qu'il convient de la décrire (de la décomposer).

<sup>15</sup> Dans ce document, nous n'abordons que la spécialisation des entités.

# Libérons-nous des contraintes !

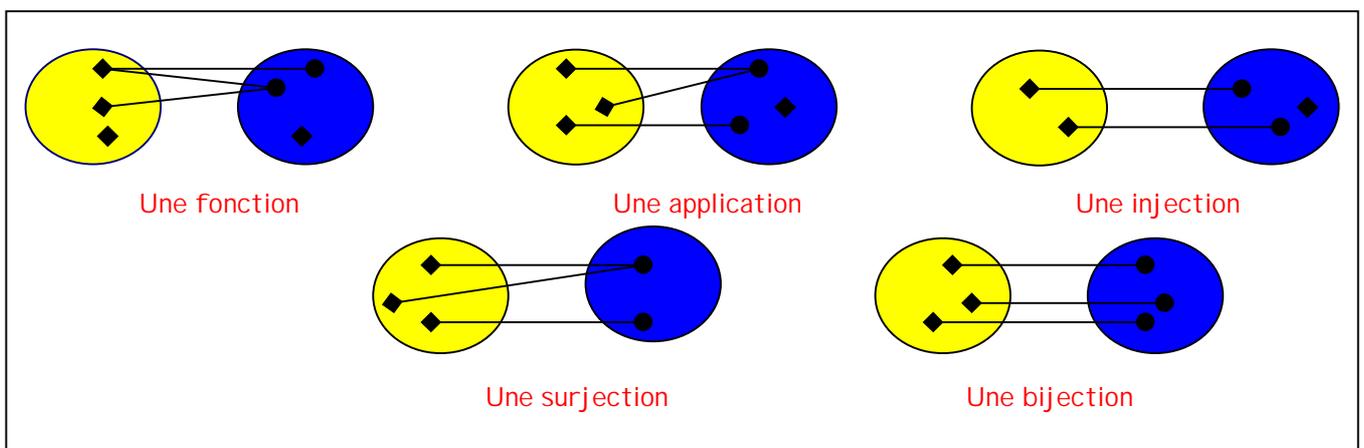
## II - Exprimer des contraintes

Le lien structurel entre deux types d'objet (par exemple un composant et son composé) peut d'abord être étudié comme une application mathématique entre deux ensembles.

### II.1 - Les contraintes applicatives

#### La référence mathématique

En mathématiques, on distingue :



Les différentes catégories mathématiques se distinguent à partir de la prise en compte des contraintes de totalité et d'unicité, dans l'ensemble de départ (en jaune) et dans l'ensemble d'arrivée (en bleu). Les questions se posent sous les formes suivantes :

- Est-ce que tout élément de l'ensemble de départ a au moins une image dans l'ensemble d'arrivée ? Est-ce qu'il en a au plus une ?
- Est-ce que tout élément de l'ensemble d'arrivée est l'image d'au moins un élément de l'ensemble de départ ? Est-ce qu'il est l'image d'au plus un élément ?

Les réponses à ces questions permettent de distinguer les catégories suivantes :

- La fonction ne connaît aucune contrainte. Ainsi, une fonction peut être indéfinie pour certaines valeurs de l'ensemble de départ.
- Dans toute application, il y a totalité et unicité au départ (TU) mais pas nécessairement à l'arrivée.
- Dans une application injective (ou injection), il y a les caractéristiques de toute application plus unicité à l'arrivée (TUU) mais pas nécessairement totalité.
- Dans une application surjective (ou surjection), il y a les caractéristiques de toute application plus totalité à l'arrivée (TUT) mais pas nécessairement unicité.
- Dans une application bijective (ou bijection), il y a les caractéristiques de toute application plus totalité et unicité à l'arrivée (TUTU), autrement dit la caractéristique d'une injection plus celle d'une surjection<sup>16</sup>.

<sup>16</sup> Injection : on ajoute U (*un jus !*). Surjection : on ajoute T (*sûreté !*). Bijection : on ajoute UT (*bijuter !* ©).

## Transposition sur le lien entre entité et propriété

### ➤ Le sens de lecture Entité vers Propriété

*A une occurrence d'entité :*

- *Correspond-il toujours au moins une occurrence de propriété, c'est-à-dire une valeur ?*

Si oui, il s'agit d'une propriété obligatoire.

Sinon, il s'agit d'une propriété facultative.

Le modèle entité-association repose sur le principe suivant : « Toute propriété est destinée à être renseignée ». Ce principe est respecté dans le schéma entité-association qui sert d'illustration, le cas GRC : toutes les propriétés sont obligatoires.

Certes, il advient qu'une propriété n'est pas renseignée, mais la seule cause éventuellement admise est que la valeur est momentanément inconnue. Par principe, la cause ne peut être l'absence de signification pour l'entité concernée. Précisément, la spécialisation est utilisée pour montrer les propriétés qui sont spécifiques à certaines catégories et n'ont pas de sens pour les autres : sourceContact pour un prospect, par exemple.

- *Correspond-il toujours une seule valeur ?*

Si oui, il s'agit d'une propriété monovaluée (ou univaluée).

Sinon, il s'agit d'une propriété multivaluée.

Le modèle entité-association reposait sur des propriétés monovaluées. Il admet désormais les propriétés multivaluées. Jusqu'à présent, cette faculté est peu utilisée dans les applications de gestion. Ainsi, dans le schéma GRC, toutes les propriétés sont monovaluées. Mais on aurait pu concevoir, par exemple, de retenir plusieurs numéros de téléphone pour un même tiers.

### ➤ Le sens de lecture Propriété vers Entité

*A une occurrence de propriété (valeur) :*

- *Correspond-il toujours au moins une occurrence d'entité ?*

La réponse est toujours oui ! C'est dans la définition même de la notion de valeur. Il n'y a pas d'électron libre dans un système d'information et, par exemple, on ne peut évidemment pas connaître un nom de tiers sans que ce soit celui d'au moins un tiers.

Cette précision est donc inutile. Elle reste implicite dans tous les modèles.

- *Correspond-il toujours une seule occurrence d'entité ?*

Si oui, il s'agit d'une propriété discriminante.

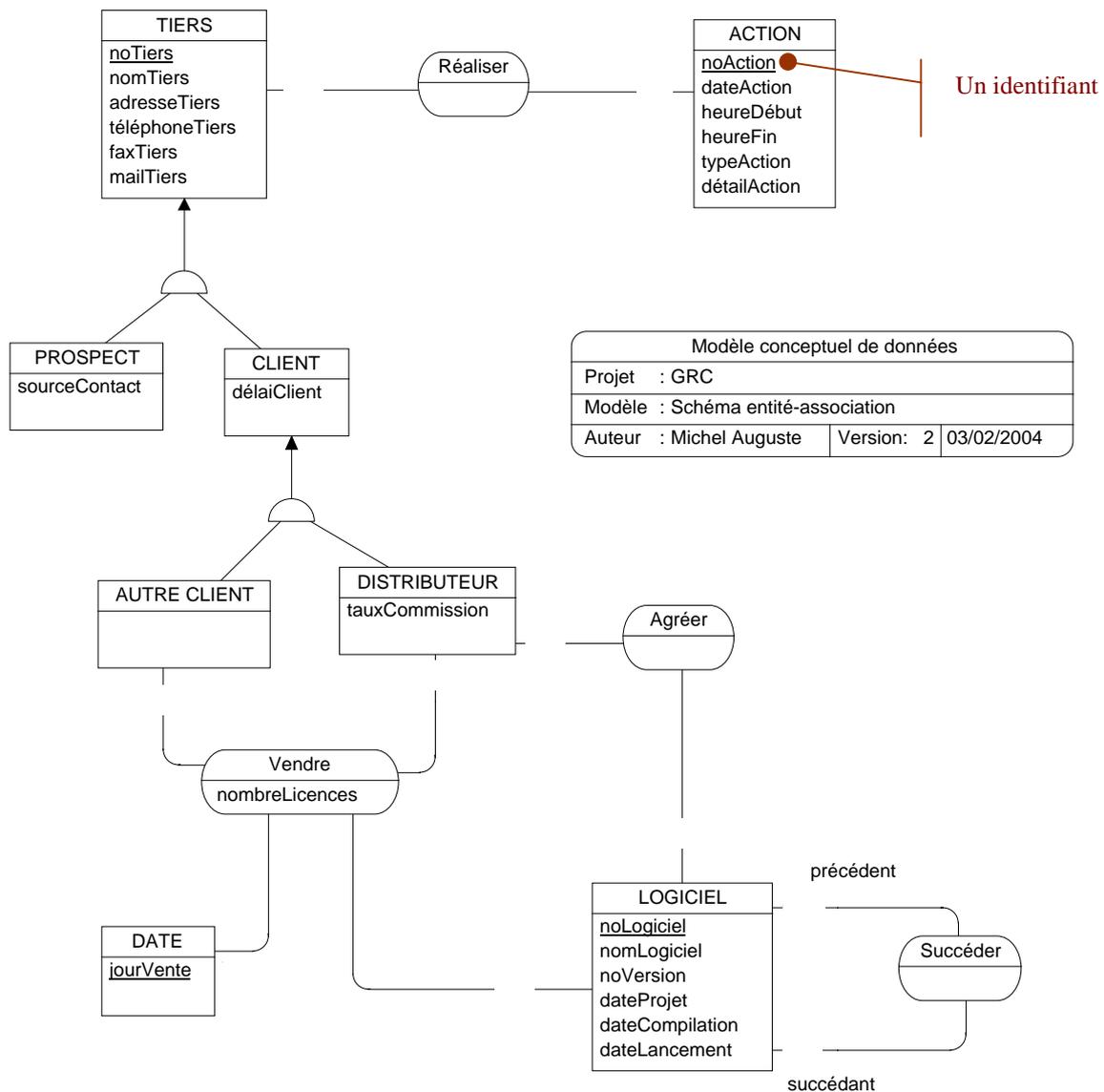
Sinon, il s'agit d'une propriété polysème.

Dans le schéma GRC, les règles de gestion impliqueront, par exemple, que noTiers soit considéré comme une propriété discriminante ; ce ne sera pas le cas d'autres propriétés, par exemple nomTiers (plusieurs tiers pouvant porter le même nom).

### ➤ Bilan : la notion d'identifiant

Dans le modèle entité-association, les propriétés sont obligatoires. Cette considération reste donc implicite. En fait, le modèle entité-association ne montre pas particulièrement les contraintes liant les propriétés à leur entité. Il privilégie une propriété par entité, propriété appelée identifiant. Sa présence est obligatoire sauf pour les entités spécialisées. En la soulignant, le modèle exprime que l'application entre cette propriété et son entité est bijective, autrement dit que la propriété est obligatoire, monovaluée et discriminante. Pour les autres propriétés, il faut se rendre dans le dictionnaire associé pour en savoir plus, par exemple pour savoir que le nom du logiciel et le n° de version permettent également d'identifier un logiciel.

Voici le schéma complété avec les seules informations données par le MEA.



Littéralement, on aurait pu écrire :

TIERS(noTiers → nomTiers, adresseTiers, téléphoneTiers, faxTiers, mailTiers)  
 ACTION(noAction → dateAction, heureDébut, heureFin, typeAction, détailAction)  
 LOGICIEL(noLogiciel → nomLogiciel, noVersion, dateProjet, dateCompilation, dateLancement)  
 DATE(jourVente)<sup>17</sup>

où la flèche se lit : « détermine », l'identifiant étant le sujet et les autres propriétés les compléments d'objets directs de ce verbe.

Vous remarquerez que les associations peuvent être porteuses de propriétés mais qu'elles ne possèdent pas d'identifiant. Cette question sera étudiée plus loin.

<sup>17</sup> Cette entité ne possède qu'une propriété qui est donc nécessairement son identifiant.

## Transposition sur le lien entre entité et association

### ➤ Le sens de lecture Entité vers Association : les cardinalités

Dans le modèle entité-association, on appelle rôle la participation d'une entité à une association et les réponses aux questions significatives des contraintes de totalité et d'unicité vont permettre de déterminer ce qu'on appelle les cardinalités.

*A une occurrence d'entité :*

- *Correspond-il toujours au moins une occurrence d'association ?*  
Si oui, on inscrit sur la « patte » une cardinalité minimale égale à 1.  
Sinon, on inscrit sur la « patte » une cardinalité minimale égale à 0.
- *Correspond-il toujours une seule occurrence d'association ?*  
Si oui, on inscrit sur la « patte » une cardinalité maximale égale à 1.  
Sinon, on inscrit sur la « patte » une cardinalité maximale égale à n.

Remarques :

- Vous noterez que l'existence d'une contrainte se traduit par le chiffre 1. Pour la cardinalité minimale, « 1 » exprime l'existence d'une contrainte de totalité. Pour la cardinalité maximale, « 1 » exprime la contrainte d'unicité.
- Si les concepteurs du modèle n'ont pas choisi une transposition exacte des concepts mathématiques, c'est pour permettre de plus grandes précisions. Ainsi, une écriture comptable mettra en jeu au moins deux comptes différents (cardinalité minimale égale à 2 sur le rôle de l'entité Écriture). Il en va de même pour... l'association qui comporte au moins deux rôles (cardinalité minimale égale à 2 sur le rôle de l'entité Association dans l'association « associer » qui la lie à l'entité Rôle<sup>18</sup> !). La cardinalité maximale peut également prendre une valeur précise. Par exemple, dans une certaine bibliothèque, une règle de gestion impose qu'un prêt ne porte pas sur plus de 5 livres (cardinalité maximale égale à 5 sur le rôle de Prêt)<sup>19</sup>.

### ➤ Le sens de lecture Association vers Entité

*A une occurrence d'association :*

- *Correspond-il toujours au moins une occurrence d'entité ?*  
La réponse est toujours oui. Dans le jargon du modèle entité-association, on dit qu'une association ne possède pas de « patte optionnelle ». Cela rejoint la notion de propriété obligatoire. Dans le modèle entité-association, tous les composants sont obligatoires mais, s'il est admis qu'une propriété non identifiante puisse ne pas être renseignée momentanément, ce n'est pas le cas des entités composant une association dont chaque occurrence ne peut exister sans les occurrences des entités correspondantes.
- *Correspond-il toujours une seule occurrence d'entité ?*  
Pour un rôle donné, la réponse est toujours oui.

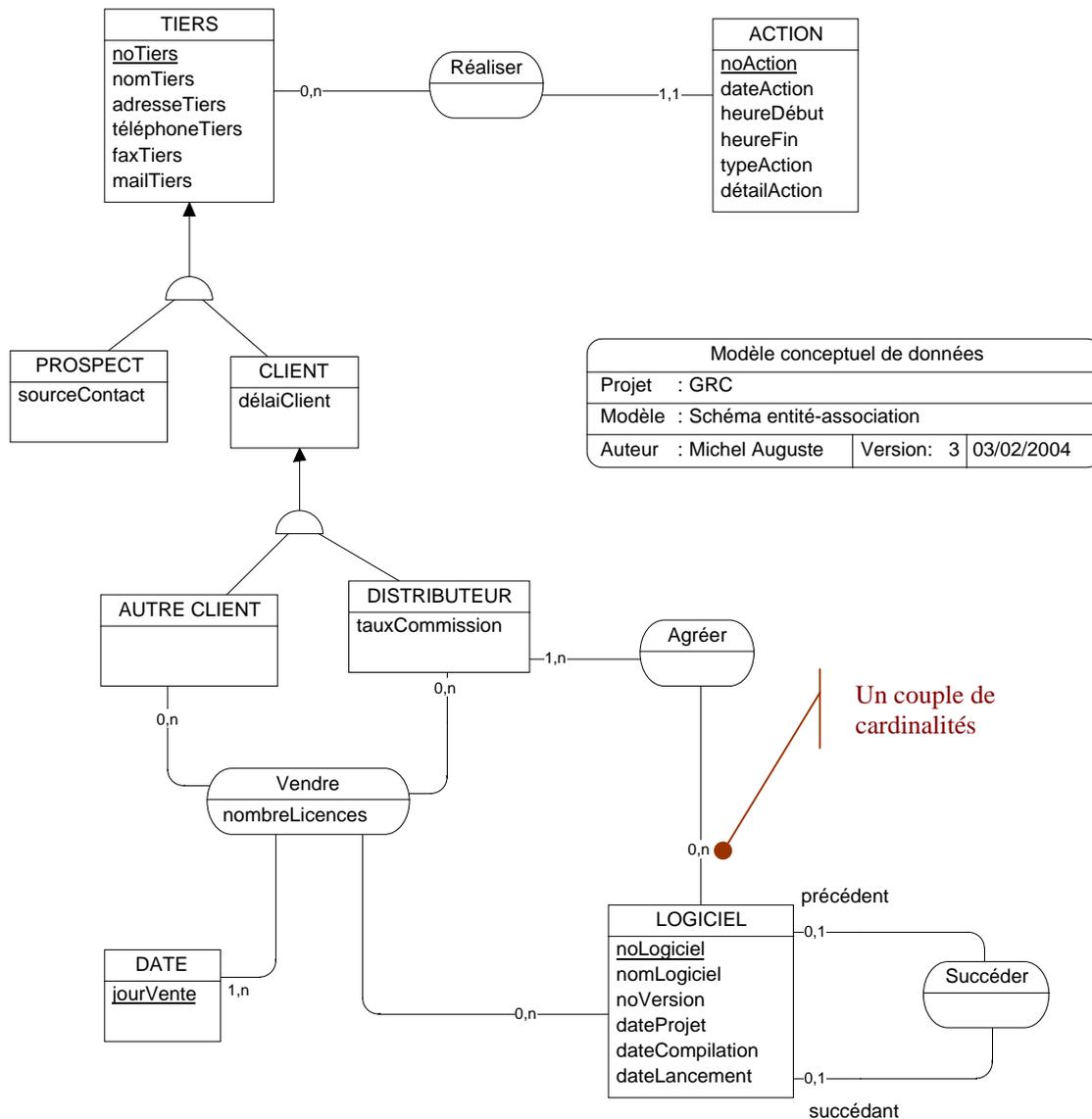
Ainsi, du côté de l'association, il n'est nul besoin de mentionner de cardinalités puisque la cardinalité minimale et la cardinalité maximale sont forcément toutes les deux égales à 1.

Page suivante, le schéma GRC est complété pour tenir compte de ces nouvelles informations.

---

<sup>18</sup> Qu'est-ce qu'on s'amuse !

<sup>19</sup> Cependant, il conviendrait de s'interroger sur la stabilité de cette règle de gestion ou plutôt de déterminer s'il s'agit d'une règle de gestion, descriptive de l'invariant du système, ou bien d'une règle « organisationnelle », modifiable.



Sur ce schéma, on peut lire désormais :

- Une action est réalisée auprès d'un et un seul tiers. Un tiers est la cible de zéro, une ou plusieurs actions.
- Un logiciel succède à au plus un logiciel. Un logiciel précède au plus un logiciel.
- Un distributeur est agréé pour au moins un logiciel. Un logiciel « agréé » zéro, un ou plusieurs distributeurs.
- Un logiciel fait l'objet de zéro, une ou plusieurs ventes (remarquez le recours au substantif pour exprimer les associations complexes). Un distributeur fait zéro, une ou plusieurs ventes. Un autre client est à l'origine de zéro, une ou plusieurs ventes. Une date (de vente) est concernée par au moins une vente<sup>20</sup>.

<sup>20</sup> La spécification est encore incomplète. Nous la complétons à la page suivante.

## ➤ Prolongement

Le nombre de « pattes », les cardinalités, la présence ou non de propriétés permettent de classer les associations. L'identification des associations dépend de ce classement.

- *Le nombre de pattes*

Les associations comportant deux rôles sont dites « binaires », les autres « n-aires ». « Réaliser », « Agréer » et « Succéder » sont des associations binaires. « Vendre » est une association n-aire (quaternaire). Dans l'association binaire « Succéder », une même entité joue les deux rôles : on parle alors d'association « réflexive ».

- *Les cardinalités*

A considérer les seules cardinalités maximales, on distingue 3 catégories d'association :

- Les associations 1 à 1 comme « Succéder ». Ce sont forcément des associations binaires.
- Les associations n vers 1, dites fonctionnelles (ou hiérarchiques quand on lit dans l'autre sens). Ce sont forcément des associations binaires. « Réaliser » est une association fonctionnelle.
- Les associations n vers n, dites maillées. « Agréer » et « Vendre » sont des associations maillées.

- *Les propriétés*

En principe<sup>21</sup>, seules les associations maillées possèdent des propriétés. C'est le cas de l'association « Vendre ». Mais la réciproque n'est pas vraie : une association maillée ne possède pas forcément de propriétés. C'est le cas de l'association « Agréer » qui sera qualifiée de « non porteuse ».

- *L'identification des associations*

- Une association maillée est identifiable par sa collection d'entités. Cela permet de préciser que :
  - Un distributeur ne peut pas vendre le même jour au même « autre client » le même logiciel. On pourrait écrire :  
[Vendre\(DISTRIBUTEUR, AUTRE CLIENT, LOGICIEL, DATE → nombreLicences\)](#).
  - Un distributeur n'est pas agréé plusieurs fois pour le même logiciel. On pourrait écrire :  
[Agréer\(LOGICIEL, DISTRIBUTEUR\)](#). Ici, l'absence de flèche indique que l'identification mobilise tous les composants.
- Une association fonctionnelle est identifiable par son rôle déterminant. Ainsi, dans « réaliser », c'est l'entité Action qui joue le rôle déterminant (cardinalités 1,1). On pourrait écrire : [Réaliser\(ACTION → TIERS\)](#).
- Une association 1 vers 1 possède deux rôles déterminants et donc autant d'identifications possibles. Dans « succéder », les rôles « succédant » et « précédent » sont deux modes d'identification candidats. On pourrait écrire :  
[Succéder\(LOGICIEL<sub>succédant</sub> ↔ LOGICIEL<sub>précédent</sub>\)](#).

---

<sup>21</sup> Dans ce document, nous nous en tiendrons à cette contrevérité.

## Transposition sur le lien entre type et sous-type

### ➤ Le sens de lecture Sous-type vers Type

A une occurrence du sous-type :

- *Correspond-il toujours au moins une occurrence du type ?*

La réponse est toujours oui.

- *Correspond-il toujours une seule occurrence du type ?*

La réponse est toujours oui.

C'est précisément ce qu'on dit en suivant la flèche : par exemple, un client est un tiers.

### ➤ Le sens de lecture Type vers Sous-type

A une occurrence du type :

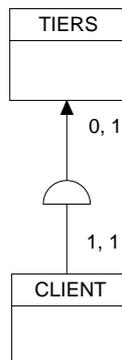
- *Correspond-il toujours au moins une occurrence du sous-type ?*

La réponse est toujours non.

- *Correspond-il toujours au plus une seule occurrence du sous-type ?*

La réponse est toujours oui.

Dans le modèle entité-association, quels que soient le type et le sous-type considérés, les « cardinalités » caractéristiques du lien entre ce type et ce sous-type sont connues.



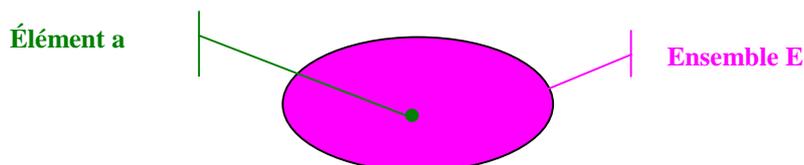
Puisqu'il ne peut en être autrement, ces « cardinalités » restent implicites. Il est inutile de les mentionner.

## II.2 – Les contraintes ensemblistes

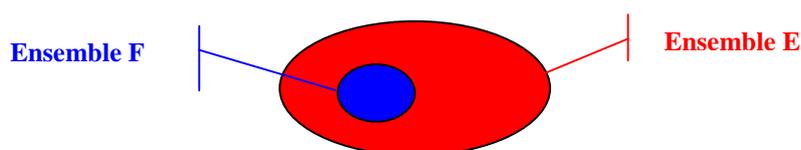
### La référence mathématique

En mathématiques, on décrit les caractéristiques des ensembles :

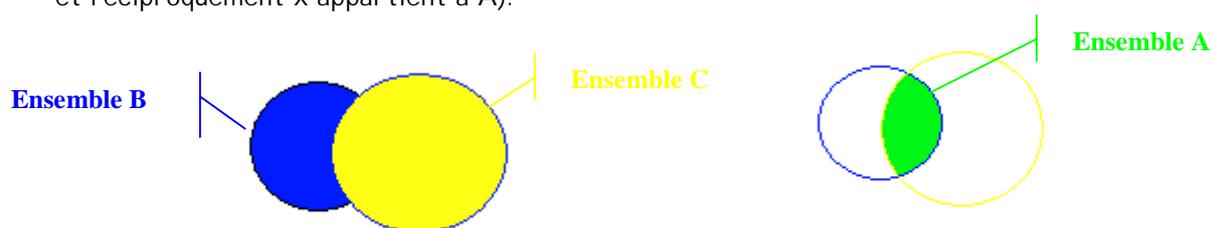
- Un ensemble  $E$  ne comporte pas deux éléments identiques.
- L'appartenance est notée :  $a \in E$  ( $a$  appartient à l'ensemble  $E$ ,  $a$  est un élément de  $E$ ).



- La non-appartenance est notée :  $b \notin E$  ( $b$  n'appartient pas à l'ensemble  $E$ ).
- Un ensemble vide est noté :  $\emptyset$ .
- L'inclusion stricte de l'ensemble  $F$  dans l'ensemble  $E$  se note :  $F \subset E$  ( $F$  est inclus dans  $E$ ).
- L'inclusion de  $F$  dans l'ensemble  $E$  se note :  $F \subseteq E$  ( $F$  est inclus ou égal à  $E$ ).



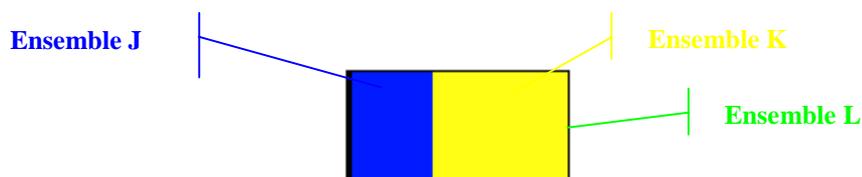
- On démontre que, si  $a \in F$  et  $F \subseteq E$ , alors  $a \in E$  (si  $a$  appartient à  $F$  et si  $F$  est inclus dans  $E$ , alors  $a$  appartient à  $E$ ).
- L'intersection est une opération qui permet de construire un nouvel ensemble à partir des éléments communs à deux ensembles. On note :  $A = B \cap C$  (l'ensemble  $A$  est l'intersection des ensembles  $B$  et  $C$ ). On peut écrire :  $x \in B$  et  $x \in C \Leftrightarrow x \in A$  ( $x$  appartient à  $B$  et à  $C$  implique et réciproquement  $x$  appartient à  $A$ ).



- Deux ensembles qui n'ont pas d'éléments communs sont dits disjoints. Si  $B$  et  $C$  sont disjoints, alors  $B \cap C = \emptyset$ .
- L'union est une opération qui permet de construire un nouvel ensemble à partir des éléments qui appartiennent à deux ensembles, soit à l'un, soit à l'autre, soit aux deux (dans ce cas, l'élément ne se trouve qu'une fois dans l'ensemble résultat puisque, par définition, un ensemble ne comporte pas d'éléments identiques). On note  $G = H \cup I$  (l'ensemble  $G$  est l'union des ensembles  $H$  et  $I$ ). On peut écrire :  $y \in H$  ou  $y \in I \Leftrightarrow y \in G$  ( $y$  appartient à  $H$  ou à  $I$  implique et réciproquement  $y$  appartient à  $G$ ).



- Des sous-ensembles d'un ensemble de référence forment une partition de celui-ci s'ils sont disjoints et si leur union donne pour résultat l'ensemble de référence. Ainsi, J et K forment une partition de L :
  - si  $J \subseteq L$  et  $K \subseteq L$  (J et K sont inclus dans L),
  - si  $J \cap K = \emptyset$  (l'intersection de J et K est vide),
  - si  $J \cup K = L$  (l'union de J et K est égale à L).



## Transposition sur la spécialisation

Les sous-types d'entité peuvent être perçus comme des sous-ensembles de leur entité générique. En dessinant une spécialisation, on a donc mentionné l'existence d'une contrainte d'inclusion. Il est inutile de le rappeler. En revanche, on peut être amené à considérer qu'il existe entre ces sous-types d'autres contraintes.

### ➤ Une contrainte d'exclusion.

C'est le cas pour les clients et les prospects. Ce sont deux sous-ensembles de Tiers. Il est exclu qu'un tiers puisse être à la fois un client et un prospect. C'est le cas également pour les distributeurs et les autres clients. Ce sont deux sous-ensembles de Client. Il est exclu qu'un client puisse être à la fois distributeur et « autre client ».

La contrainte d'exclusion se note à l'aide du symbole **X**.

### ➤ Une contrainte de totalité (ou couverture).

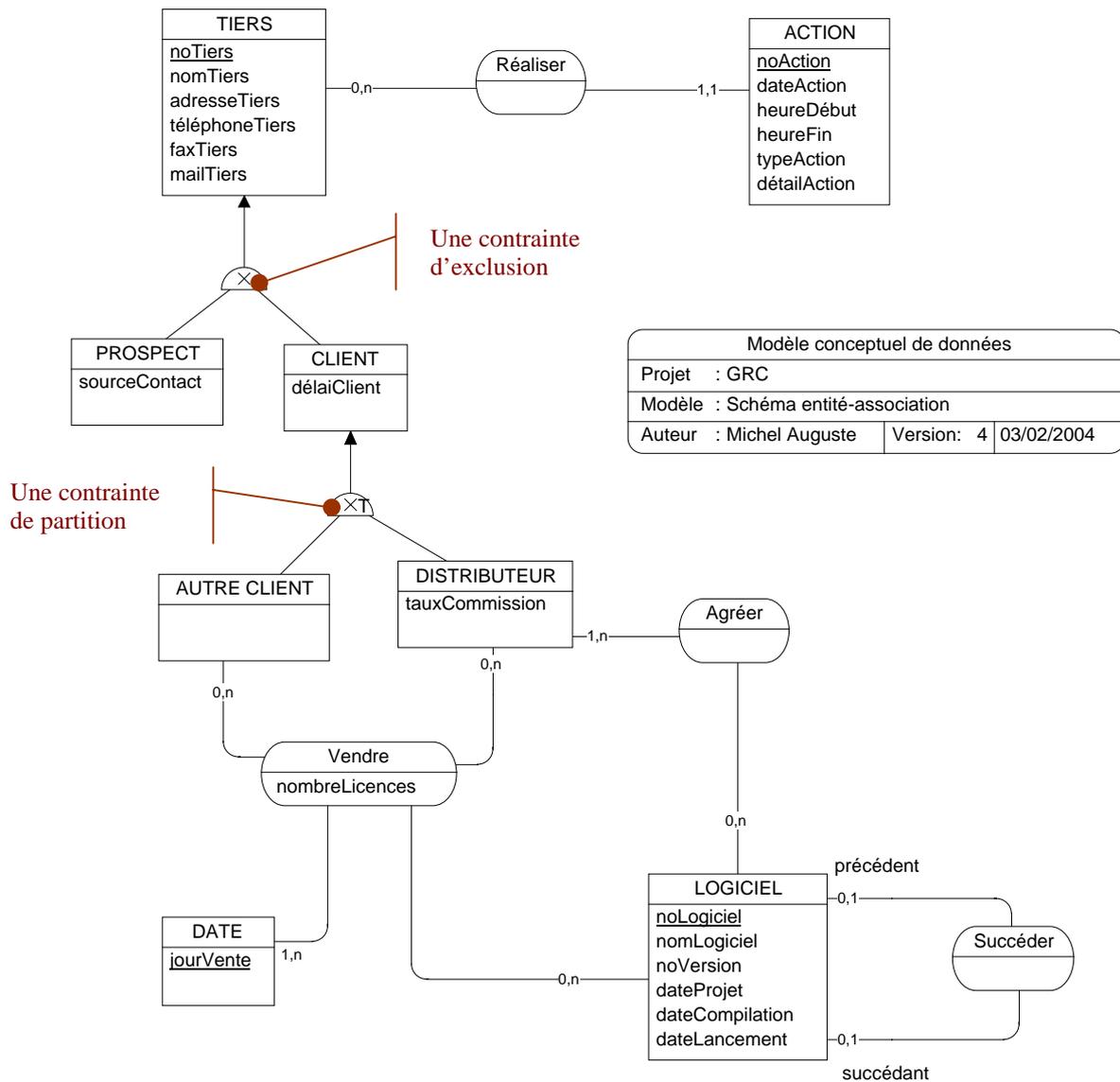
C'est le cas des distributeurs et des autres clients. Il n'y a pas d'autres clients que les distributeurs et les autres clients. En revanche, ce n'est pas le cas pour les prospects et les clients. Certes, le schéma ne présente pas d'autres catégories mais l'énoncé du thème précisait : « Parmi les tiers, on distingue notamment les prospects et les clients ». L'adverbe « notamment » nous suggère qu'il existe d'autres tiers que les prospects et les clients.

La contrainte de totalité se note à l'aide du symbole **T**.

### ➤ Une contrainte de partition.

C'est le cas des distributeurs et des autres clients pour lesquels il y a bien à la fois une contrainte d'exclusion et une contrainte de totalité. Comme précisément, la présence d'une double contrainte (exclusion et totalité) implique et réciproquement l'existence d'une contrainte de partition, il n'est pas nécessaire de disposer d'un symbole particulier. Nous choisirons cette option : **XT** représentera donc une contrainte de partition. Sachez cependant que certains modèles particularisent cette contrainte à l'aide du symbole  $\oplus$ .

Voici le nouveau schéma GRC prenant en compte ces considérations.



Ici, c'est dans la demi-lune représentative d'une spécialisation que les informations sont placées :

- **X**, contrainte d'exclusion, sur la spécialisation de Tiers en Prospect ou Client.
- **XT**, contrainte de partition, sur la spécialisation de Client en Distributeur ou Autre Client.

## Transposition sur les associations

➤ **Un rôle est un sous-ensemble. Une association est un ensemble de n-uplets<sup>22</sup>.**

Pour bien appréhender ce qui suit, il vous faut d'abord prendre conscience qu'un rôle est assimilable à un sous-ensemble.

Lorsqu'on place les cardinalités  $0, n$  sur le rôle de Tiers dans l'association « Réaliser », on dit en fait que les tiers cibles d'actions constituent un sous-ensemble des Tiers. Nous allons choisir d'écrire le fait de cette manière :

Réaliser[TIERS]  $\subseteq$  TIERS

ce qu'on peut lire ainsi : « L'ensemble obtenu en projetant « Réaliser » sur le rôle TIERS est inclus dans l'ensemble TIERS ».

De la même manière, en plaçant les cardinalités  $0, 1$  sur le rôle « succédant » de l'entité LOGICIEL dans l'association « Succéder », on dit que les logiciels successeurs constituent un sous-ensemble de l'ensemble des logiciels. Nous allons choisir d'écrire le fait de cette manière :

Succéder[LOGICIEL<sub>succédant</sub>]  $\subseteq$  LOGICIEL

ce qu'on peut lire ainsi : « L'ensemble obtenu en projetant « Succéder » sur le rôle LOGICIEL succédant est inclus dans l'ensemble LOGICIEL ».

Nous venons de prendre deux exemples pour lesquels la cardinalité minimale égale à 0 implique un lien d'inclusion. Mais, et ceci est à considérer attentivement, lorsque la cardinalité minimale est égale à 1, le lien est un lien d'égalité. Ainsi, on voit que :

Réaliser[ACTION] = ACTION (l'ensemble obtenu en projetant « Réaliser » sur le rôle ACTION est égal à l'ensemble ACTION).

Agréer[DISTRIBUTEUR] = DISTRIBUTEUR

Pourquoi réclamer votre attention sur ce point ? Regardez la « patte » de DISTRIBUTEUR dans l'association « Agréer » : nous venons de dire que tous les distributeurs sont « sur cette patte ». Regardez maintenant la « patte » de DISTRIBUTEUR dans l'association « Vendre ». La cardinalité minimale égale à 0 nous incite à écrire :

Vendre[DISTRIBUTEUR]  $\subseteq$  DISTRIBUTEUR

Rapprochons les deux faits :

Agréer[DISTRIBUTEUR] = DISTRIBUTEUR

Vendre[DISTRIBUTEUR]  $\subseteq$  DISTRIBUTEUR

Vous paraît-il utile de préciser

Vendre[DISTRIBUTEUR]  $\subseteq$  Agréer[DISTRIBUTEUR] ?

Non, évidemment, pour vous, cela va de soi que « certains » est forcément inclus dans « tous ». C'est pourtant une erreur assez fréquente dans les sujets portant sur le « modèle étendu » (y compris des sujets de CAPET option D !<sup>23</sup>).

Toujours est-il que le fait de considérer les rôles comme des sous-ensembles peut impliquer l'expression de contraintes. Pour nous faire comprendre, nous allons quelques temps quitter le domaine de la GRC, qui ne s'y prête pas pour l'instant.

---

<sup>22</sup> Un ensemble de couples pour une association binaire, de triplets pour une association ternaire, de n-uplets pour une association n-aire.

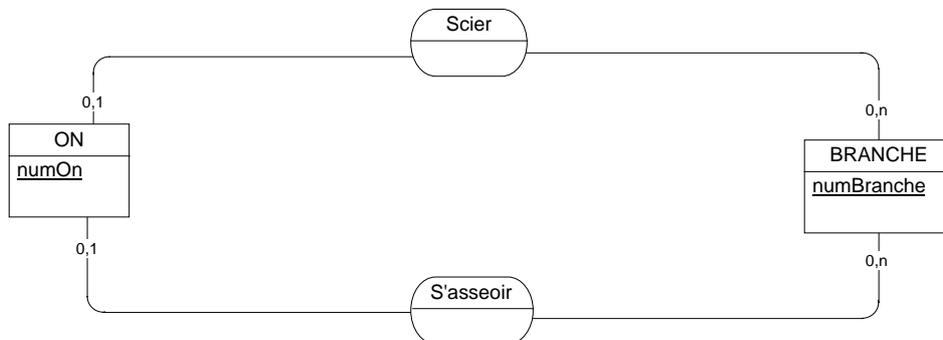
<sup>23</sup> Il est à souhaiter que les candidats soient moins... étendus que le modèle ! ☺

## 1<sup>er</sup> exemple : La contrainte d'exclusion

« On ne scie pas la branche sur laquelle on est assis. »<sup>24</sup>

La variété d'interprétations possibles naît d'une petite tricherie : nous admettrons que les deux « on » que contient la phrase sont des pronoms indéfinis au sens strict, donc ne renvoient pas forcément à la même personne.

Présentons d'abord la situation générale que nous admettrons pour hypothèse :

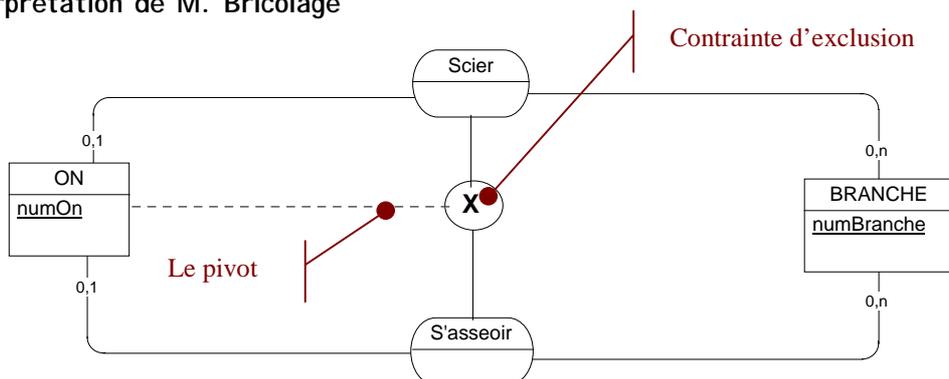


On scie au plus une branche ; une branche est sciée par zéro, un ou plusieurs « on »<sup>25</sup>.

On est assis sur au plus une branche ; une branche supporte zéro, un ou plusieurs « on ».

Montrons maintenant les diverses interprétations du proverbe.

### ➤ L'interprétation de M. Bricolage



Comme pour les sous-types, la pastille X exprime une contrainte d'exclusion. Mais, dans le modèle entité-association, on a choisi de situer la contrainte entre les associations concernées quelle que soit la signification précise de la contrainte à exprimer. C'est le pivot de la contrainte, représenté par un trait en pointillés, qui lève l'ambiguïté. Ici, on pourrait écrire :  $Scier[ON] \cap S'asseoir[ON] = \emptyset$ .

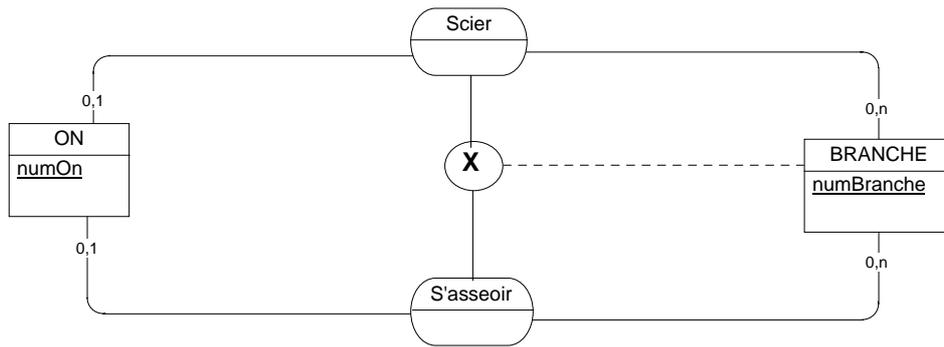
L'avis de M. Bricolage est tout simplement qu'on ne scie pas assis<sup>26</sup> (ah, non !). Vous lui rétorquerez sans doute : « Mais, M. Bricolage, vous permettez de scier une branche supportant d'autres « on ». Vous passiez sous silence ce problème incident (de scie) ! ». M. Bricolage vous répondra : « Je n'ai pas été formé pour ceci, mais savez-vous que, si six scies scient six cyprès, six cent six scies scient six cent six cyprès ? ».

<sup>24</sup> Exemple emprunté à M. Henri Habrias.

<sup>25</sup> Scier la même branche à plusieurs ? Aurait-on à faire à une bande de « on » ?

<sup>26</sup> Là, nous n'en disons... pas assez. Il convient de préciser : « Pas assis... sur une branche ! ». En effet, dans le schéma, rien n'interdit aux scieurs d'être assis sur autre chose qu'une branche. Vous nous pardonnerez cette supercherie.

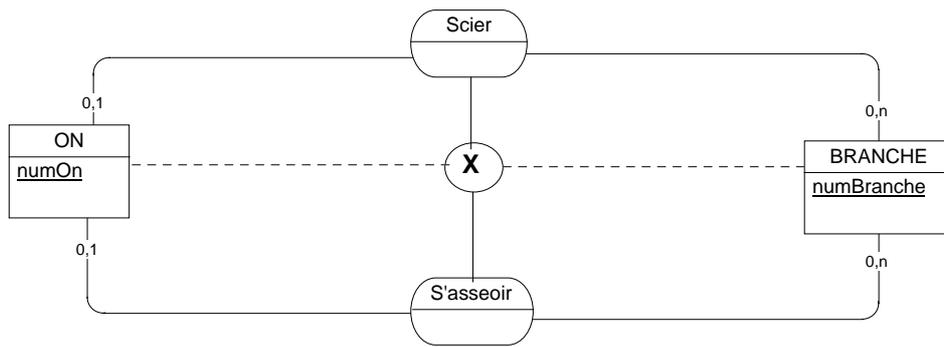
➤ L'interprétation de M. Sécurité



Ici on pourrait écrire :  $\text{Scier}[\text{BRANCHE}] \cap \text{S'asseoir}[\text{BRANCHE}] = \emptyset$ .

L'opinion de M. Sécurité est qu'on doit s'assurer que la branche qu'on scie ne supporte pas « quelqu'un » (ou quelques « on », y compris soi-même). Voilà un bon présage, précis et sage, pour le pré-scilage mais qui n'interdit pas à un scieur de scier assis pourvu qu'il soit assis sur une autre branche. Ce qui ne sied guère à M. Bricolage...

➤ L'interprétation de M. Piaget



Ici, la contrainte ne porte pas sur l'un des rôles mais sur les couples associés. En effet, chaque association est ici assimilable à un ensemble de couples. Les associations « Scier » et « S'asseoir » sont donc des ensembles de couples ON-BRANCHE et, à ce titre, elles sont comparables. On pourrait écrire :  $\text{Scier} \cap \text{S'asseoir} = \emptyset$ .<sup>27</sup>

L'idée de M. Piaget est qu'on ne va pas scier la branche sur laquelle on est soi-même assis<sup>28</sup>. C'est concis mais rien n'interdit à notre scieur d'être assis sur une autre branche<sup>29</sup>, rien n'interdit à notre scieur de scier une branche supportant ne serait-ce qu'un petit « on »<sup>30</sup>, donc rien n'interdit au premier « on » venu de scier la branche sur laquelle notre scieur est assis<sup>31</sup>.

<sup>27</sup> Dans ce document, nous n'envisagerons pas les cas où le pivot peut rester implicite. Sachez que c'est précisément le cas dans cet exemple.

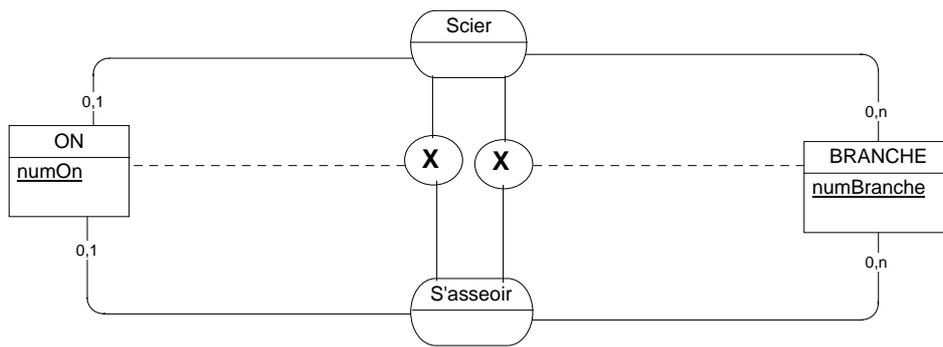
<sup>28</sup> Personne n'est assez « on » pour ça. C'est bien là l'interprétation la plus commune du proverbe.

<sup>29</sup> La leçon de M. Bricolage n'est donc pas assi... milée.

<sup>30</sup> Un assis mineur, quoi !

<sup>31</sup> Au revoir, mon scieur !

➤ L'interprétation de M. Yanne<sup>32</sup>



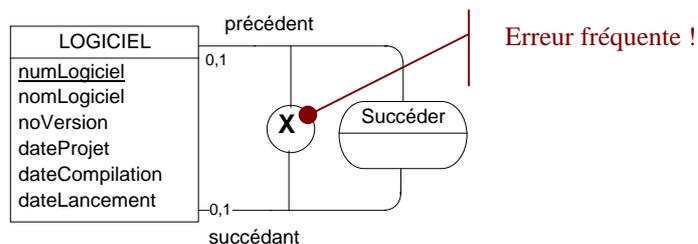
Ici, nous avons deux contraintes d'exclusion, chacune portant sur l'un des rôles.

M. Yanne déclare : « Tout le monde, il est beau, tout le monde, il est gentil. »<sup>33</sup>. Donc pas question de scier assis, pas question de scier une branche supportant « quelqu'on »...<sup>34</sup>

Nous ferons remarquer aussi (ainsi qu'aux scieurs et aux sciés) que cette interprétation intègre toutes les autres. *See you later !*

Transposition sur le cas GRC

On peut être tenté de placer une contrainte d'exclusion sur l'association réflexive « succéder ».



Remarquez que, dans ce cas, il faut bien accrocher la contrainte aux « pattes ». Mais ce n'est pas sans conséquence. Interrogeons-nous, en effet, sur le sens de cette contrainte. Pour déterminer la signification d'une contrainte, il faut, en principe, en placer le pivot. Ici, franchement, est-ce bien nécessaire ? Tout est dit, puisque la contrainte est accrochée aux rôles. Il n'y a qu'une seule interprétation possible : un logiciel jouant le rôle de précédent ne peut jouer le rôle de succédant. On pourrait écrire :

$$\text{Succéder}[\text{LOGICIEL}_{\text{précédent}}] \cap \text{Succéder}[\text{LOGICIEL}_{\text{succédant}}] = \emptyset.$$

Or, cette contrainte n'est pas celle que nous voulions mentionner. Pis, elle est erronée : si le logiciel n° 15 succède au logiciel n° 4 et le logiciel n° 33 succède au logiciel n° 15, nous voyons bien que, dans des couples différents, le logiciel n° 15 remplit alternativement les deux rôles. Ce que nous voulions dire est qu'un logiciel ne peut jouer les deux rôles dans le même couple, ce que nous pourrions écrire ainsi :  $\forall S \in \text{Succéder} : S.\text{LOGICIEL}_{\text{précédent}} \neq S.\text{LOGICIEL}_{\text{succédant}}$ .

Remarquez que la contrainte ainsi écrite nous amène à considérer la composition de chaque n-uplet et non à comparer des sous-ensembles dans leur globalité. En conséquence, la contrainte à mentionner n'est pas une contrainte ensembliste. Nous y reviendrons plus loin.

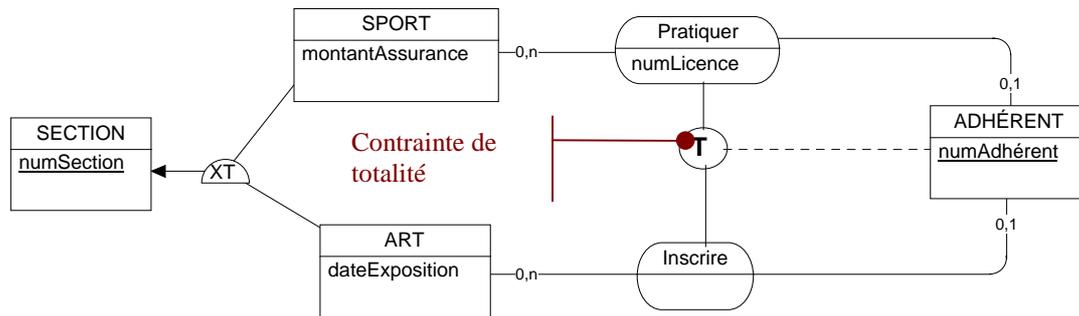
<sup>32</sup> En guise d'hommage !

<sup>33</sup> C'est une « scie » bien connue.

<sup>34</sup> Reconnaissons que ce n'est pas supportable !

## 2<sup>e</sup> exemple : La contrainte de totalité

Un exemple commenté nous suffira ici (inspiré du cas Tarmel, CAPET externe 1999).



Il existe des sections parmi lesquelles on distingue deux catégories : Sport et Art. Ces deux sous-ensembles forment une partition de l'ensemble des sections.

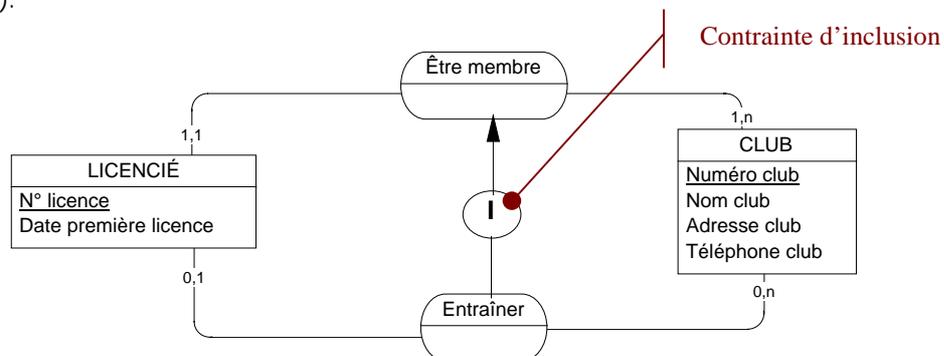
Il existe des adhérents qui pratiquent un sport ou s'inscrivent à une section d'art ou qui font les deux mais au moins l'un des deux.

Si la contrainte de totalité n'est pas précisée, il pourrait exister des adhérents n'exerçant aucune activité, en contradiction avec la règle de gestion énoncée : il n'est pas obligatoire de pratiquer un Sport (cardinalité minimale égale à 0 sur le rôle d'Adhérent dans l'association « pratiquer »), il n'est pas obligatoire de s'inscrire à une section Art (cardinalité minimale égale à 0 sur le rôle d'Adhérent dans l'association « inscrire ») mais il est obligatoire d'exercer l'une des deux activités (contrainte de totalité liant les rôles d'Adhérent).

On peut écrire :  $\text{Pratiquer}[\text{ADHÉRENT}] \cup \text{Pratiquer}[\text{ADHÉRENT}] = \text{ADHÉRENT}$ .

## 3<sup>e</sup> exemple : la contrainte d'inclusion à source unique

L'exemple qui suit est extrait et adapté d'un sujet de BTS développeur d'applications (Triathlète, Nouméa 2000).

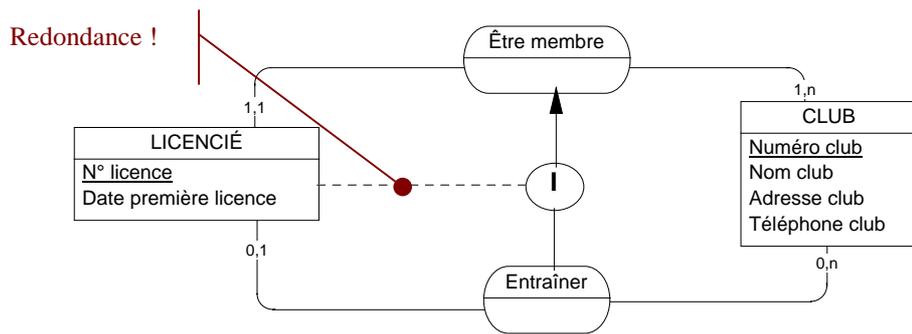


On y lit que tous les licenciés sont membres d'un club. Un club accueille au moins un licencié. Certains licenciés sont entraînés dans un club. Un club entraîne zéro, un ou plusieurs licenciés. Une contrainte d'inclusion (pastille I) a été placée entre les associations « Entraîner » et « Être membre ». Comme il s'agit d'une contrainte d'inclusion, le lien est orienté de la source « Entraîner » vers la cible « Être membre ».

A notre connaissance, la contrainte d'inclusion est la seule contrainte orientée, ce qui est logique puisque c'est la seule contrainte où les éléments liés par la contrainte n'ont pas le même statut : à la source, l'inclus, au bout de la flèche, l'incluant. On ne peut pas en dire autant ni d'une contrainte d'exclusion ni d'une contrainte de totalité<sup>35</sup>

<sup>35</sup> Nous revenons sur ces questions plus loin dans la présentation du méta-modèle.

Dans le schéma précédent, le pivot n'a pas été mentionné, à dessein. Nous allons étudier maintenant les interprétations possibles de la contrainte selon le pivot choisi.

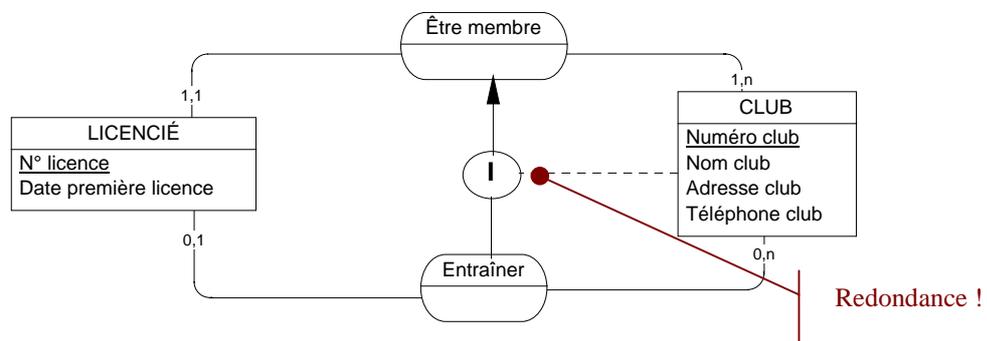


Ici, le schéma « précise » que tous les licenciés entraînés sont des licenciés membres. Or, cette précision est inutile puisque tous les licenciés sont membres ! Retenez bien qu'une contrainte d'inclusion entre deux rôles dont le rôle cible est doté d'une cardinalité minimale égale à 1 est tout simplement redondante !

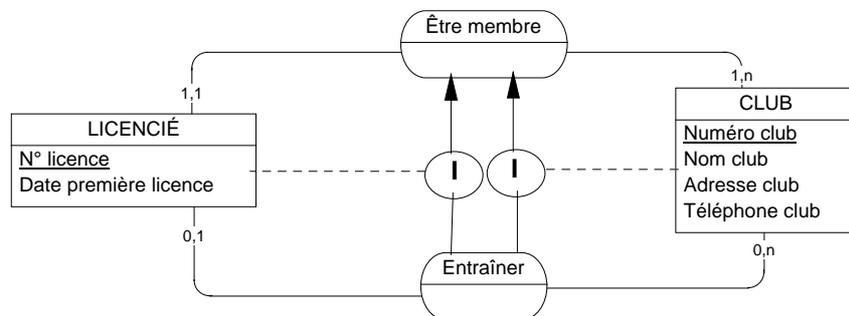
### Formulation

- Nous avons :
1. Être membre[LICENCIÉ] = LICENCIÉ (cardinalité minimale = 1)
  2. Entraîner[LICENCIÉ]  $\subseteq$  LICENCIÉ (cardinalité minimale = 0)
  3. Entraîner[LICENCIÉ]  $\subseteq$  Être membre[LICENCIÉ]  
(contrainte d'inclusion qui ne dit rien de plus que nous ne sachions déjà)

Il en va donc de même lorsqu'on fait porter la contrainte d'inclusion sur les rôles de Club.

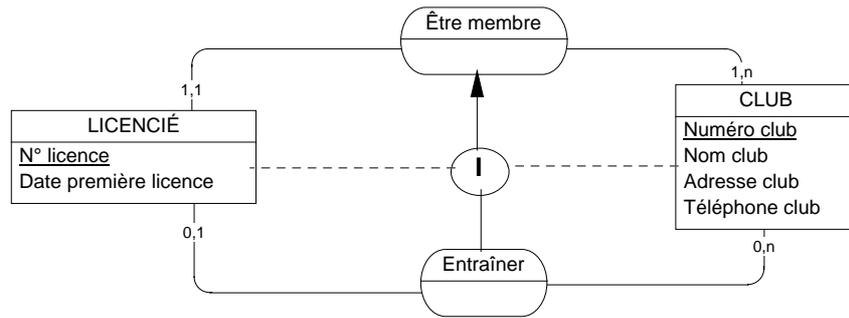


Évidemment, la fusion des schémas ne change rien à l'affaire.



Ce qu'on veut exprimer, c'est qu'un licencié est entraîné dans un club dont il est membre. La contrainte porte sur les couples. Pour l'indiquer, le pivot relie les deux entités<sup>36</sup>.

<sup>36</sup> Dans ce cas, d'ailleurs, le pivot peut rester implicite.

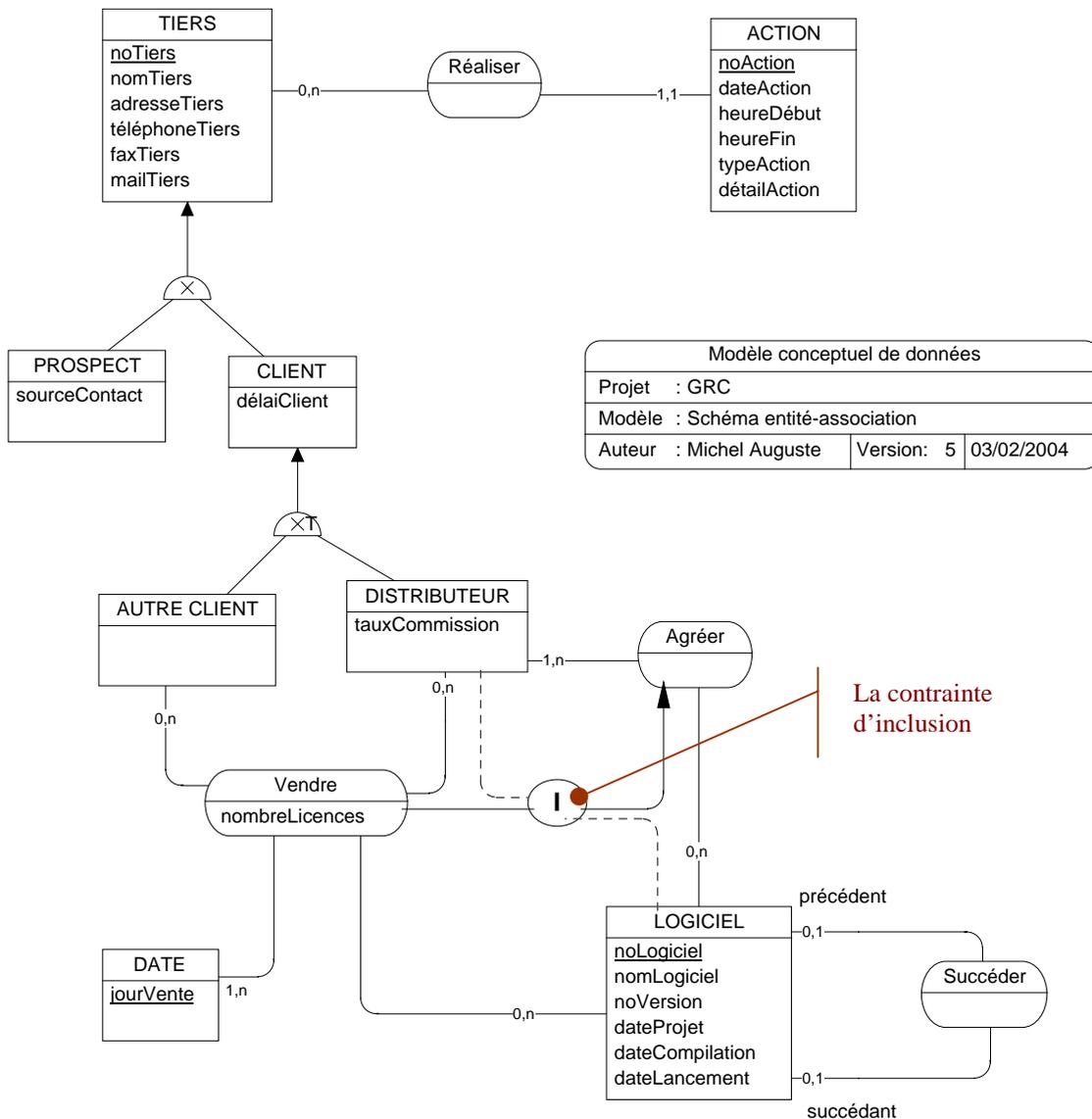


On pourrait écrire :  $\text{Entraîner} \subseteq \text{Être membre}$ .

Nous allons appliquer cette nouvelle connaissance au cas GRC, pour exprimer que les distributeurs vendent des logiciels pour lesquels ils ont été agréés.

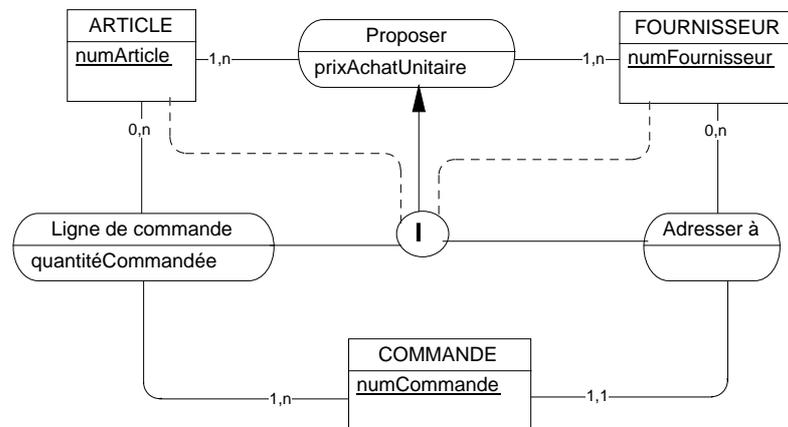
La contrainte porte sur le couple d'entités associées par « Agréer » et qui fait partie de la collection de « Vendre ». On pourrait écrire :

$\text{Vendre}[\text{DISTRIBUTEUR}, \text{LOGICIEL}] \subseteq \text{Agréer}$ .



#### 4<sup>e</sup> exemple : la contrainte d'inclusion à source composée

Examinons l'exemple suivant :



Le schéma exprime notamment que :

- Un article est proposé par au moins un fournisseur. Un fournisseur propose au moins un article. Le prix d'achat dépend de l'article et du fournisseur.
- Une commande est adressée à un fournisseur. Un fournisseur reçoit zéro, une ou plusieurs commandes.
- Une commande porte sur au moins un article. Un article fait l'objet de zéro, une ou plusieurs commandes. La quantité commandée dépend de la commande et de l'article.

La contrainte d'inclusion signifie que les articles concernés par une commande adressée à un fournisseur font partie des articles proposés par ce fournisseur.

La source est composée et, pour transcrire littéralement la contrainte, nous utiliserons une nouvelle opération que nous nommerons la jointure et que nous noterons  $\otimes$ .

$(\text{Ligne de commande} \otimes \text{Adresser à})[\text{ARTICLE}, \text{FOURNISSEUR}]$   
 $\subseteq \text{Proposer}[\text{ARTICLE}, \text{FOURNISSEUR}]$

ce qui se lit : « La jointure de « Ligne de commande » et de « Adresser à »<sup>37</sup>, projetée sur ARTICLE et FOURNISSEUR, est incluse dans « Proposer » projetée sur les mêmes composants ».

Nous verrons plus loin qu'une erreur fréquente consiste à « oublier » des « pattes » dans la source, pattes sans lesquelles la contrainte n'est pas formellement interprétable.

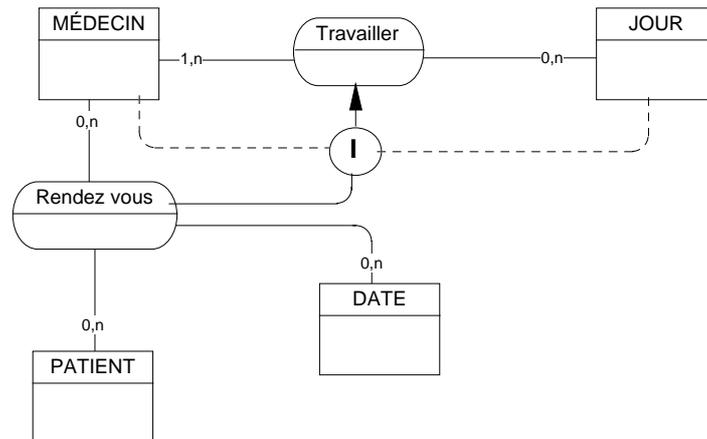
#### La contrainte d'égalité

Il nous semble que la contrainte d'égalité ne possède pas de caractéristique propre (voir méta-modèle).

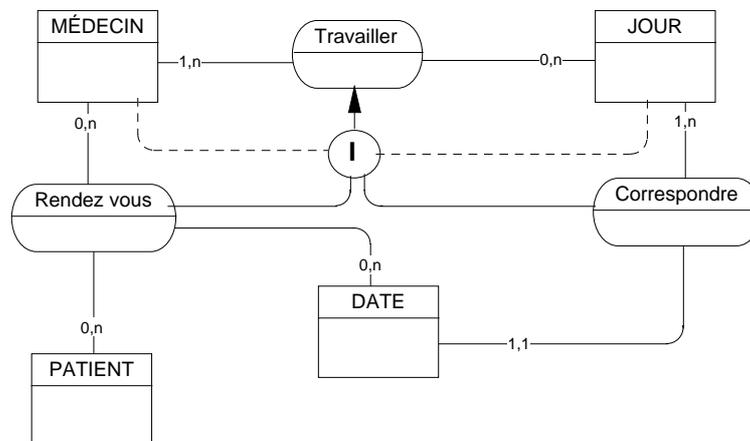
<sup>37</sup> Dans ce document, nous laissons de côté le problème que peut poser la multiplicité des chemins de jointure possibles. Dans notre exemple, il n'y a pas d'ambiguïté : « Ligne de commande » et « Adresser à » n'ont qu'un composant portant sur la même référence (par abus de langage, nous pourrions dire « qu'un composant commun »), c'est COMMANDE. C'est donc par ce chemin que s'effectue la « jointure ».

## Quelques contre-exemples (erreurs les plus fréquentes)

- Contre-exemple n° 1 : Erreur de référence (cas Les Tilleuls, EDC CAPET D externe 1998)



Formellement, l'entité JOUR ne fait pas partie de la collection de l'association Rendez vous : elle ne saurait faire partie du pivot d'une contrainte d'inclusion dont Rendez vous serait la source. Qu'est-ce qui a conduit certains étudiants préparant le CAPET à proposer cela ? Le sujet précisait que les médecins du cabinet travaillaient certains jours de la semaine : le domaine de JOUR est donc {'lundi', 'mardi', 'mercredi', etc.} Ces étudiants voulaient préciser que les rendez-vous pris par un médecin avec un patient l'étaient forcément à une date non chômée par ce médecin. Ils ont construit cette contrainte d'inclusion sans appréhender le fait que le référentiel de DATE n'est pas celui de JOUR, même s'il y a évidemment un rapport entre les deux. D'autres ont aperçu la difficulté et l'ont contournée ainsi :



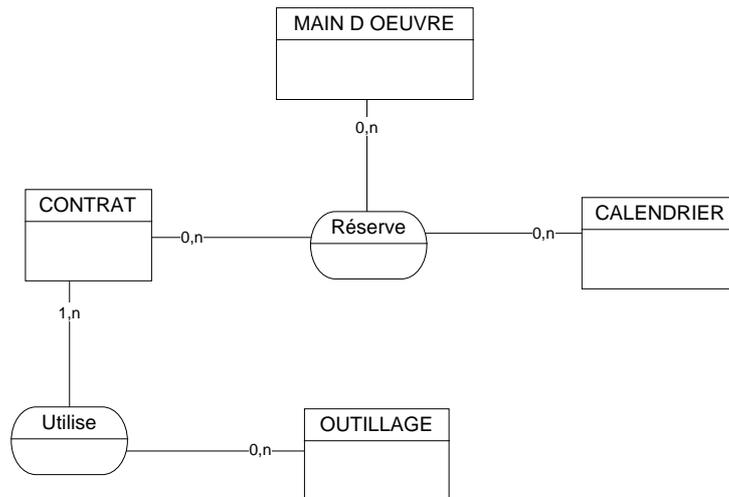
Dans le contexte du sujet, ce n'est pas très élégant mais ce qui est présenté est valide d'un point de vue formel :

$(\text{Rendez vous} \otimes \text{Correspondre})[\text{MÉDECIN}, \text{JOUR}] \subseteq \text{Travailler}$

Les entités MÉDECIN et JOUR font bien partie de l'union des collections des associations jointes.

- Contre-exemple n° 2 : Tautologie (Cas Outillage, sujet d'EST, CAPET D externe 2002)

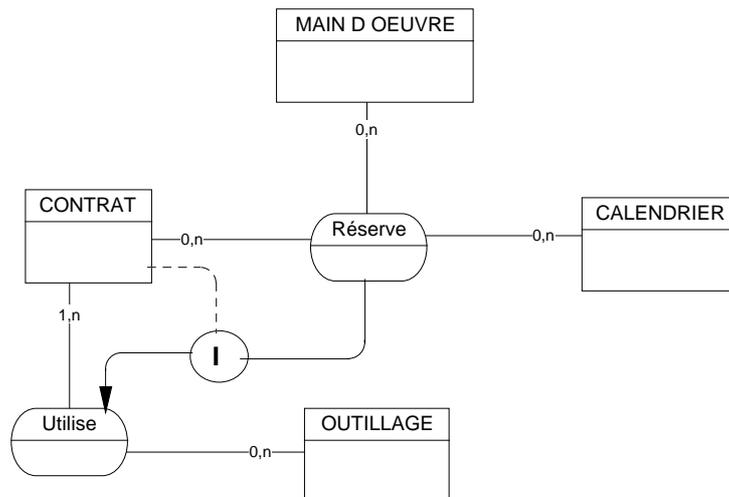
Extrait du schéma contenu dans le sujet :



Question posée : Exprimez ces contraintes par des notations de votre choix dans le schéma conceptuel.

Parmi les contraintes à exprimer, on trouve : « Un contrat ne peut utiliser<sup>38</sup> une main d'œuvre que si un outillage est loué ».

Le corrigé publié dans le rapport de jury contient :



Une contrainte d'inclusion qui est, nous l'avons vu, complètement inutile !

La cardinalité minimale sur le rôle de CONTRAT dans l'association Réserve est égale à 0 :

$\text{Réserve}[\text{CONTRAT}] \subseteq \text{CONTRAT}$ .

La cardinalité minimale sur le rôle de CONTRAT dans l'association Utilise est égale à 1 :

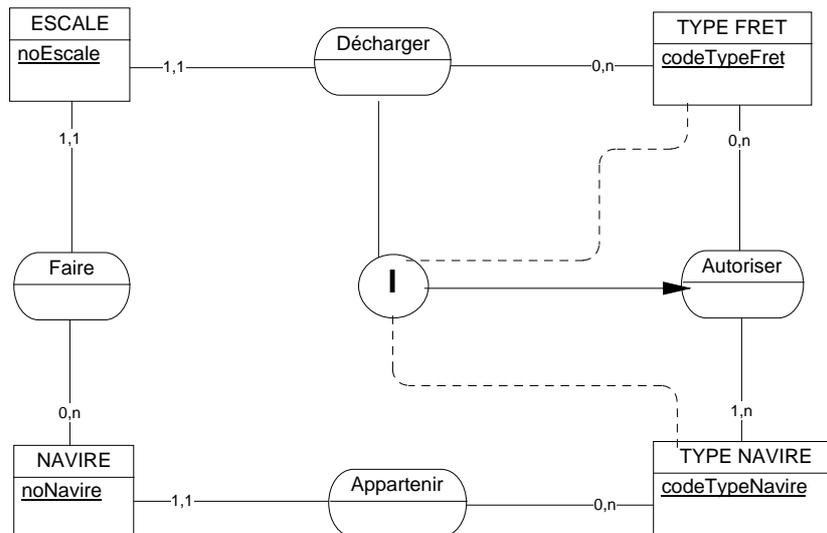
$\text{Utilise}[\text{CONTRAT}] = \text{CONTRAT}$ .

$\text{Réserve}[\text{CONTRAT}] \subseteq \text{Utilise}[\text{CONTRAT}]$  est donc déjà établi.

<sup>38</sup> Nous avons conservé l'intitulé original de la question qui emploie pour parler du lien entre CONTRAT et MAIN D'ŒUVRE le verbe employé dans le schéma pour l'autre lien.

- Contre-exemple n° 3 : Incomplétude (inspiré de CAEN-Ouistreham, BTS DA Nouméa 2001)

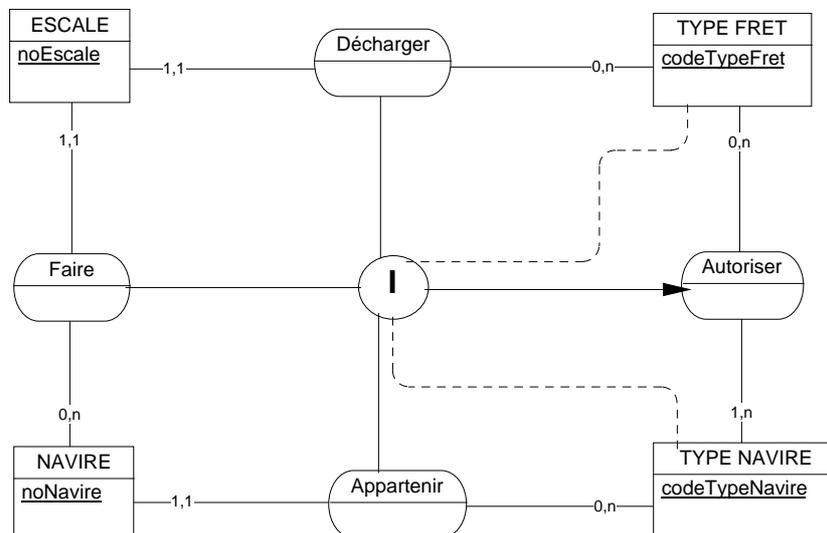
Extrait de la solution produite par plusieurs étudiants :



Il s'agissait de contrôler qu'un navire qui fait escale « chez nous » pour y décharger tel type de fret est bien autorisé à transporter ce type de fret.

L'erreur commise ici n'est qu'une variante de la première mais elle est beaucoup plus fréquente (on la trouve même dans certaines documentations d'ateliers de génie logiciel !). Formellement, TYPE NAVIRE ne fait pas partie de la collection d'entités de l'association Décharger. Celle-ci ne saurait donc être la source unique d'une contrainte d'inclusion comportant TYPE NAVIRE en son pivot.

Il faut faire les « jointures » nécessaires :

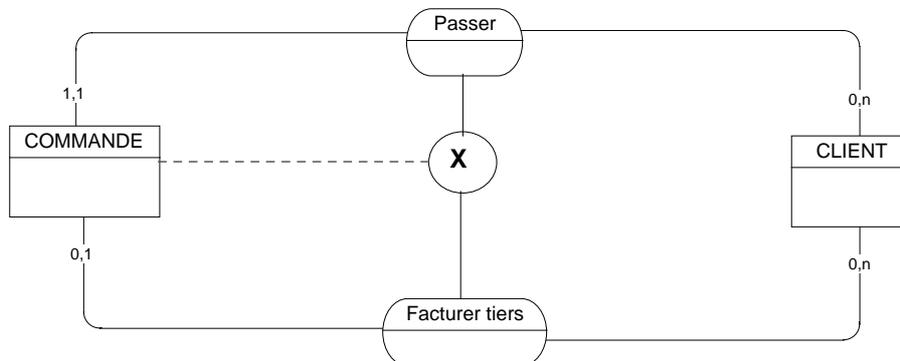


où la contrainte d'inclusion exprime :

$(\text{Décharger} \otimes \text{Faire} \otimes \text{Appartenir})[\text{TYPE NAVIRE}, \text{TYPE FRET}] \subseteq \text{Autoriser}$

- Contre-exemple n° 4 : Contradiction (inspiré de l'exemple documentant WinDesign)

Une commande est passée par un client mais certaines sont facturées à d'autres clients.



Formellement, l'écriture de cette contrainte pour laquelle le pivot est commande ne tient pas :

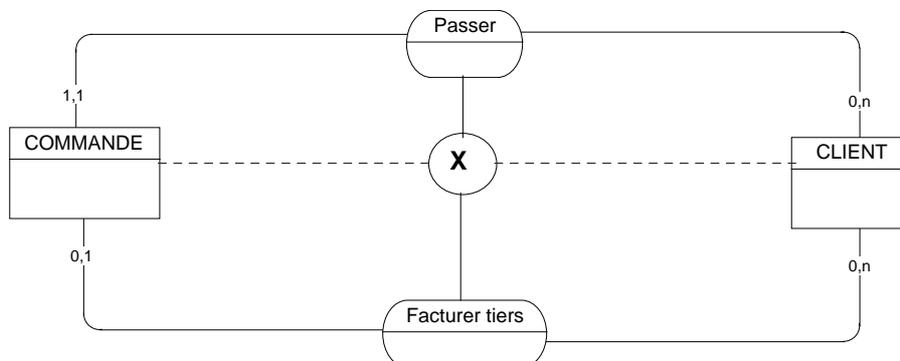
Cardinalité 1 sur le premier rôle  $\Leftrightarrow$   $\text{Passer}[\text{COMMANDE}] = \text{COMMANDE}$

Cardinalité 0 sur le deuxième rôle  $\Leftrightarrow$   $\text{Facturer tiers}[\text{COMMANDE}] \subseteq \text{COMMANDE}$

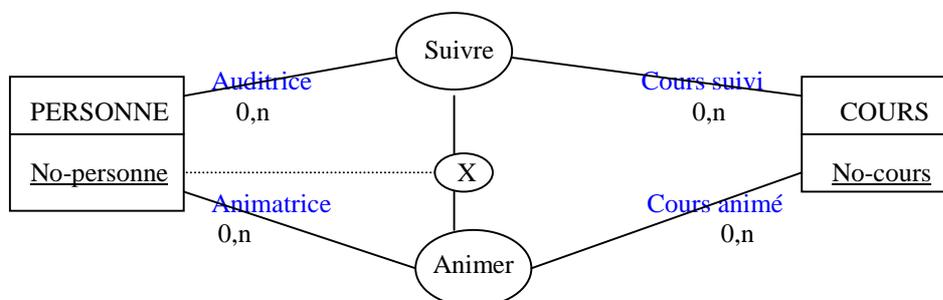
D'où il vient :  $\text{Facturer tiers}[\text{COMMANDE}] \subseteq \text{Passer}[\text{COMMANDE}]$

Or, on sait bien qu'il ne peut y avoir en même temps inclusion et exclusion !

L'expression « naturelle » de la contrainte exprime bien que celle-ci porte sur le couple : la commande peut être facturée à d'autres clients... C'est donc à juste titre que la documentation de WinDesign propose :



Au niveau de l'apprentissage, l'utilisation des rôles en lecture (interprétation) et en écriture (conception) est une mesure préventive qui nous semble efficace. En voici un exemple :



La contrainte se lit ainsi : les personnes auditrices ne peuvent être des personnes animatrices et réciproquement. Remarquons que cet énoncé ne fait pas intervenir le concept de cours, ce qui justifie le choix du pivot.

## II.3 – Les relations binaires

### La référence mathématique

En mathématiques, on utilise la notion de relation binaire pour définir les relations d'équivalence et les relations d'ordre.

- Soit  $E$  un ensemble. On appelle relation binaire sur  $E$  toute application  $R$  de l'ensemble produit  $E \times E$  (ou  $E^2$ ) dans l'ensemble à deux éléments {'vrai', 'faux'} (ou  $\{0, 1\}$ ).

Notation :  $x$  et  $y$  appartenant à  $E$ , le composé est noté  $xRy$  ou  $R(x, y)$  et on écrit :

- soit «  $xRy$  est vrai » ou plus simplement «  $xRy$  » ou encore «  $R(x, y) = 1$  » ;
  - soit «  $xRy$  est faux » ou «  $R(x, y) = 0$  ».
- A la base des définitions des relations d'équivalence et des relations d'ordre, il y a la réflexivité, la symétrie, l'antisymétrie et la transitivité. Pour les besoins de la cause, nous complétons la panoplie ci-dessous. Soit  $E$  un ensemble muni d'une relation binaire  $R$ .

*Peut-on être en relation avec soi-même ?*

- Réflexivité.

La relation  $R$  est dite réflexive si quel que soit  $x$  appartenant à  $E$ ,  $xRx$  est vrai.

$$\forall x \in E, xRx$$

- Antiréflexivité.

La relation  $R$  est dite antiréflexive si quel que soit  $x$  appartenant à  $E$ ,  $xRx$  est faux.

$$\forall x \in E, \text{non } xRx$$

*Peut-on jouer les deux rôles ?*

- Symétrie.

La relation  $R$  est dite symétrique si quels que soient  $x$  et  $y$  appartenant à  $E$ , «  $xRy$  est vrai » implique et réciproquement «  $yRx$  est vrai ».

$$\forall (x, y) \in E^2, xRy \Leftrightarrow yRx$$

- Antisymétrie.

La relation  $R$  est dite antisymétrique si quels que soient  $x$  et  $y$  appartenant à  $E$ , «  $xRy$  est vrai » et «  $yRx$  est vrai » implique «  $x$  est égal à  $y$  ».

$$\forall (x, y) \in E^2, xRy \text{ et } yRx \Rightarrow x = y$$

Une autre formulation est possible : la relation  $R$  est dite antisymétrique si quels que soient  $x$  et  $y$  appartenant à  $E$ , avec  $x$  différent de  $y$ , «  $xRy$  est vrai » implique «  $yRx$  est faux ».

$$\forall (x, y) \in E^2 \text{ et } x \neq y, xRy \Rightarrow \text{non } yRx$$

- Asymétrie.

La relation  $R$  est dite asymétrique si quels que soient  $x$  et  $y$  appartenant à  $E$ , «  $xRy$  est vrai » implique «  $xRx$  est faux » et «  $yRx$  est faux ».

$$\forall (x, y) \in E^2, xRy \Rightarrow (\text{non } xRx) \text{ et } (\text{non } yRx)$$

*Entraîne-t-on les autres ? ...*

- Transitivité.

La relation  $R$  est dite transitive si quels que soient  $x$ ,  $y$  et  $z$  appartenant à  $E$ , «  $xRy$  est vrai » et «  $yRz$  est vrai » impliquent «  $xRz$  est vrai ».

$$\forall (x, y, z) \in E^3, xRy \text{ et } yRz \Rightarrow xRz$$

- Intransitivité.

La relation  $R$  est dite intransitive si quels que soient  $x$ ,  $y$  et  $z$  appartenant à  $E$ , «  $xRy$  est vrai » et «  $yRz$  est vrai » impliquent «  $xRz$  est faux ».

$$\forall (x, y, z) \in E^3, xRy \text{ et } yRz \Rightarrow \text{non } xRz$$

... Avec retour à l'envoyeur ?

- Cyclicité.

La relation R est dite cyclique si quels que soient x, y et z appartenant à E, « xRy est vrai » et « yRz est vrai » impliquent « zRx est vrai ».

$$\forall (x, y, z) \in E^3, xRy \text{ et } yRz \Rightarrow zRx$$

- Acyclicité.

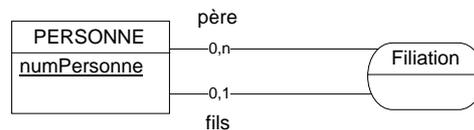
La relation R est dite acyclique si quels que soient x, y et z appartenant à E, « xRy est vrai » et « yRz est vrai » impliquent « zRx est faux ».

$$\forall (x, y, z) \in E^3, xRy \text{ et } yRz \Rightarrow \text{non } zRx$$

## Transposition sur les associations « réflexives »

➤ À une association réflexive, correspond une relation binaire.

Soit E une entité. L'association réflexive A(E<sub>1</sub>, E<sub>2</sub>) où E<sub>1</sub> et E<sub>2</sub> sont les deux rôles joués par l'entité E dans l'association A est transposable en une relation binaire : E (l'entité) est l'ensemble de référence, l'association A évoque la relation R qui associe à tout couple de E×E la valeur 'vrai' (le couple est une occurrence de A) ou la valeur 'faux' (le couple n'est pas une occurrence de A).



Soit PERSONNE = {P1, P2, P3, P4, P5, P6} (l'ensemble PERSONNE compte 6 occurrences) et Filiation = {(P1, P2), (P2, P4), (P5, P3)} où, dans les couples cités, le premier élément joue le rôle de père et le deuxième le rôle de fils (l'ensemble Filiation compte 3 couples).

Nous pouvons considérer la relation binaire « *père\_de* » et écrire, par exemple, « P1*père\_de*P2 est vrai » ou encore « P1*père\_de*P4 est faux ».

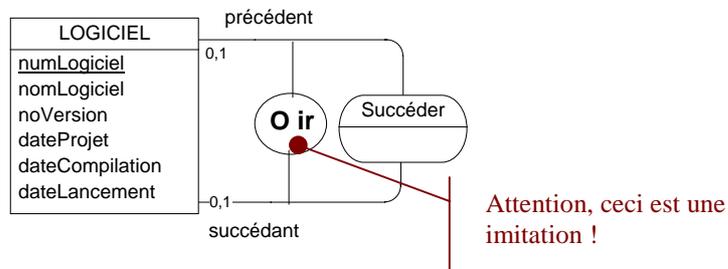
➤ L'expression des contraintes.

Dans l'exemple « Filiation », nous voudrions pouvoir dire que :

- On ne peut pas être son propre père (antiréflexivité).
- Si une première personne est le père d'une deuxième personne, alors cette dernière ne peut être le père de la première (puisque l'on a antiréflexivité, cette contrainte est assimilable à la propriété d'asymétrie).
- Si une première personne est le père d'une deuxième personne et cette deuxième personne le père d'une troisième personne, alors la première ne peut être le père de la troisième (intransitivité).
- Si une première personne est le père d'une deuxième personne et cette deuxième personne le père d'une troisième personne, alors la troisième ne peut être le père de la première (acyclicité).

En fait, le modèle entité-association n'intègre pas l'expression de ces genres de contraintes même parmi ses diverses extensions. Un modèle américain<sup>39</sup> s'attarde sur ces contraintes où elles sont nommées *ring constraints*. Nous nous en inspirons ci-dessous à propos du cas GRC.

Dans le cas GRC, nous avons rencontré une association « réflexive », l'association « Succéder ». Précédemment, l'expression d'une contrainte nous a posé problème. Il s'agissait de montrer qu'un logiciel ne peut succéder à lui-même. La propriété correspondante est... l'antiréflexivité<sup>40</sup>. En ORM, cela s'appelle *irreflexive constraint* notée **O ir**.



En fait, ici, il faudrait mentionner les autres contraintes comme l'asymétrie. Pourtant nous nous en tiendrons là. En effet, il ne s'agit pas pour nous de proposer de nouvelles extensions au MEA, bien assez riche déjà, mais de donner la possibilité à tout un chacun de maîtriser la notion de contrainte ensembliste en évitant de la confondre, dans les associations dites « réflexives », avec les propriétés des relations binaires correspondantes.

<sup>39</sup> Il s'agit de la méthode ORM, continuateur de la méthode NIAM dont le modèle de données, le modèle relationnel binaire, a été « popularisé » en France par Henri Habrias.

<sup>40</sup> Une association réflexive dotée d'une contrainte d'antiréflexivité. Paradoxal, non ?

# Objets, vos papiers !

## III - L'identification des composés<sup>41</sup>

Le MEA, avons-nous dit, repose, par principe, sur deux niveaux de composition qui nous permettent de distinguer deux types de composé : l'entité et l'association. Contrairement aux modèles à objets qui en confient la responsabilité au « système », le MEA reste préoccupé par l'identification des composés<sup>42</sup>. Nous avons vu que, par principe, :

- Toute entité possède une propriété identifiante.
- Toute association est identifiable par tout ou partie de sa collection d'entités.

Pour refléter la réalité d'un système, ces principes (identification absolue des entités, composition limitée à deux niveaux) sont quelque peu restrictifs. Ils ont été transgressés depuis longtemps dans les modèles anglo-saxons. Les nouveaux concepts proposés font partie du vocabulaire du métier aujourd'hui et c'est pourquoi nous les présenterons rapidement dans leur forme originelle.

### III.1 - L'interprétation anglo-saxonne

#### L'entité faible (*weak entity*)

##### *Cas typique*

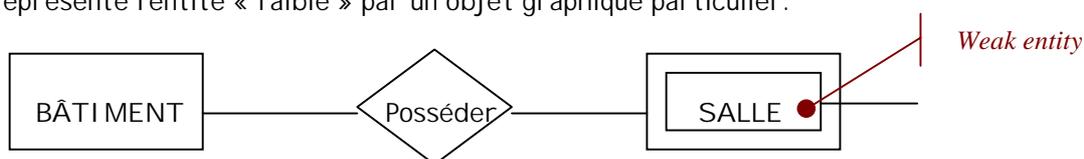
Dans un lycée, une salle est caractérisée par un numéro qui l'identifie au sein du bâtiment auquel elle appartient.

##### *Problématique*

Si nous considérons les bâtiments et les salles, nous avons affaire à des entités « emboîtées ». On parle aussi, en France, « d'entités gigognes ». Du point de vue du mode d'identification, l'entité SALLE ne nous fournit pas un mode d'identification intrinsèque. Pour identifier complètement une salle, il faut savoir à quel bâtiment elle appartient. On dit que SALLE est une entité « faible », dont l'entité « forte » est BÂTIMENT.

##### *Solution anglo-saxonne*

On représente l'entité « faible » par un objet graphique particulier.



Dans le modèle anglo-saxon, l'entité est représentée par un rectangle couché sans cartouche. L'entité faible est alors représentée par un rectangle dans un rectangle (ce qui favorise la visualisation de l'emboîtement). Le seul lien accroché au rectangle extérieur est celui qui lie l'entité faible à son entité forte. Si l'entité faible a des liens avec d'autres entités, ces liens s'accrocheront sur le rectangle intérieur. Dans l'exemple

<sup>41</sup> Nous reprenons ici des thèmes que nous avons développés par ailleurs (voir annexe du document « Vraies et fausses ternaires »).

<sup>42</sup> Celle des composants ne pose de problème.

ci-dessus, le lien dont le départ est figuré pourrait exprimer la participation de l'entité SALLE à l'association « Occuper ». <sup>43</sup>

### L'agrégat (*aggregation*)<sup>44</sup>

#### *Cas typique*

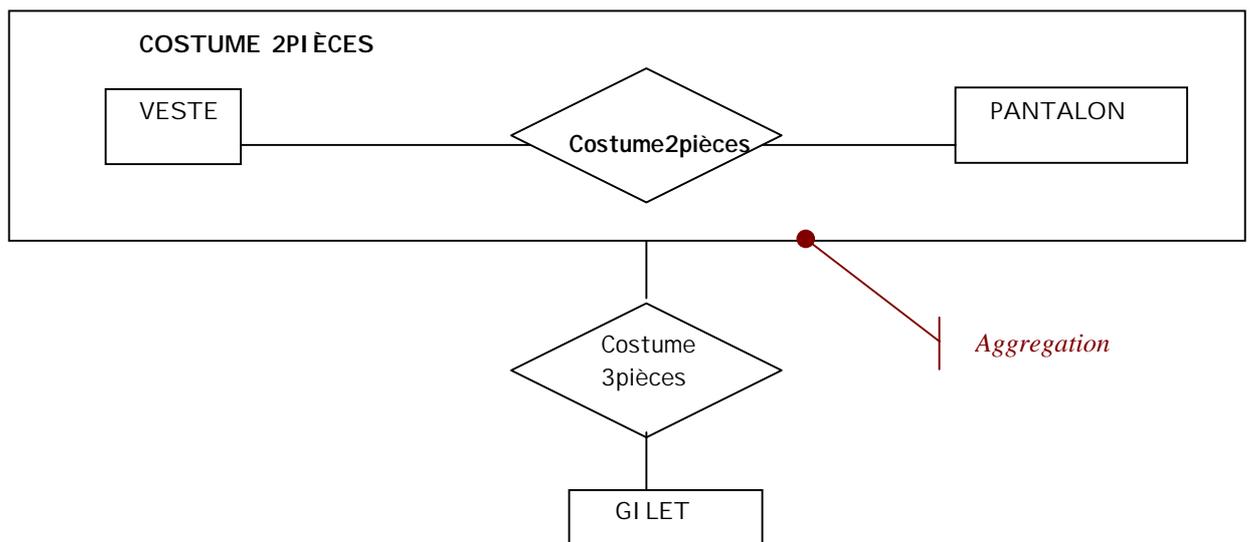
Dans une entreprise de confection, on fabrique des costumes deux-pièces à partir de vestes et de pantalons existants. On fabrique aussi des costumes trois-pièces à partir de costumes deux-pièces existants auxquels s'ajoutent les gilets.

#### *Problématique*

Si nous considérons les costumes comme des associations, nous avons d'une part une association binaire (costume deux-pièces), d'autre part une association ternaire (costume trois-pièces). Mais les deux associations côte à côte ne transcrivent pas la réalité et l'on pourra faire n'importe quel costume trois-pièces, pourvu qu'il soit composé d'une veste, d'un pantalon et d'un gilet. Il faudrait pouvoir construire l'association « ternaire » sur l'association binaire. En s'en donnant les moyens, on considérera que « costume deux-pièces » est un agrégat (ou une entité agrégative ou encore une pseudo-entité).

#### *Solution anglo-saxonne*

On représente l'agrégat par un objet graphique particulier.



## III.2 - La solution française

En France, on n'a pas choisi d'objet graphique particulier pour représenter ces notions. Pour satisfaire les besoins énoncés, mais aussi des besoins posés par des situations plus complexes, que nous qualifierons d'« hybrides », la plupart des auteurs d'extensions au MEA s'est accordé sur la mise en avant du mode d'identification des composés, donnant ainsi naissance aux nouveaux concepts suivants :

<sup>43</sup> Nous n'avons pas représenté les propriétés et les « cardinalités », qui s'expriment de façon différente du modèle français.

<sup>44</sup> Dans les années 1980, on parlait aussi de personnalisation (cf. ACISIOME).

### L'identifiant relatif.

Exemple : Dans le premier cas présenté ci-dessus, le n° de salle identifie la salle dans le bâtiment auquel elle appartient. Le n° de salle n'est pas un identifiant absolu : c'est un identifiant relatif.

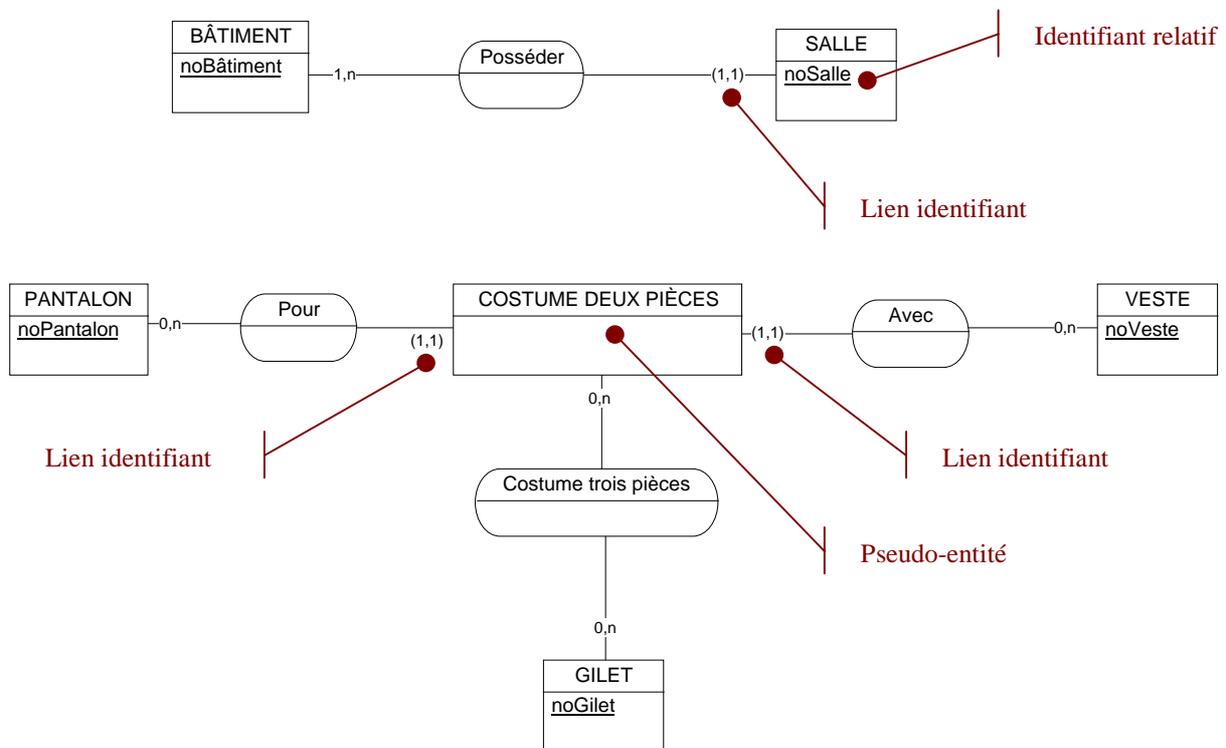
L'identifiant relatif n'a pas de représentation particulière. C'est la présence ou non de liens identifiants pour l'entité considérée qui nous permet de savoir si un identifiant est relatif ou absolu.

### Le lien identifiant.

Exemple : Dans le premier cas présenté ci-dessus, le lien entre Salle et Bâtiment (posséder) participe à l'identification de l'entité Salle. On dit qu'il s'agit d'un lien identifiant.

Le lien identifiant possède une représentation particulière qui diffère selon les auteurs. Dans ce qui suit, nous avons opté pour celle qui est la plus usitée en BTS informatique de gestion (celle du logiciel Power\*AMC).

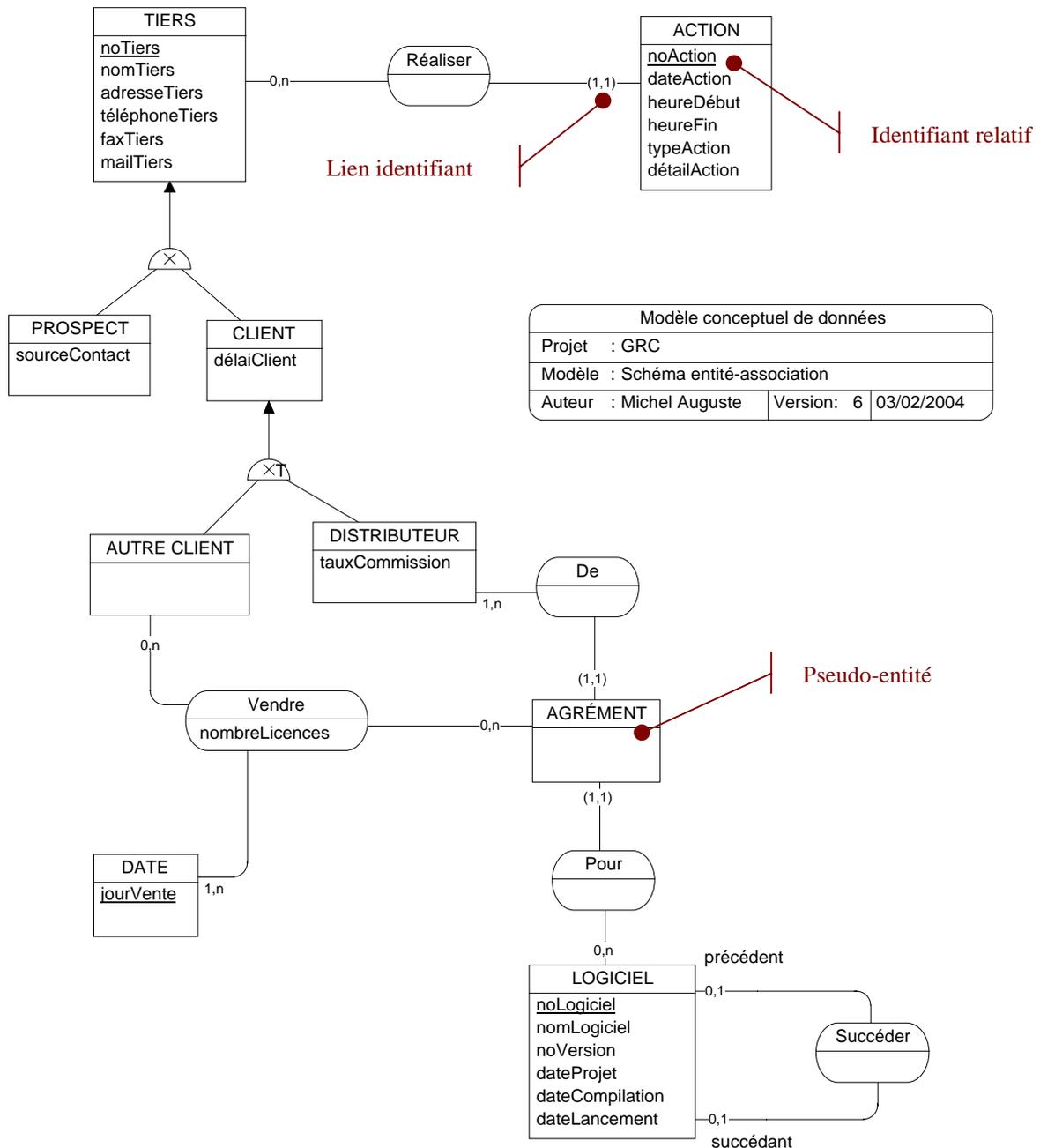
Voici les représentations « à la française » des deux cas d'école précédents.



Cela nous amène à réviser les règles concernant l'identification des entités :

- Dans le cas général, une entité possède un identifiant absolu.
- Si une entité est concernée par au moins un lien identifiant et possède une propriété soulignée, alors cette dernière est un identifiant relatif.
- Une entité générique qui ne possède pas de propriété soulignée doit être concernée par au moins deux liens identifiants.
- Nous savons par ailleurs qu'une entité spécialisée peut ne pas posséder d'identifiant.

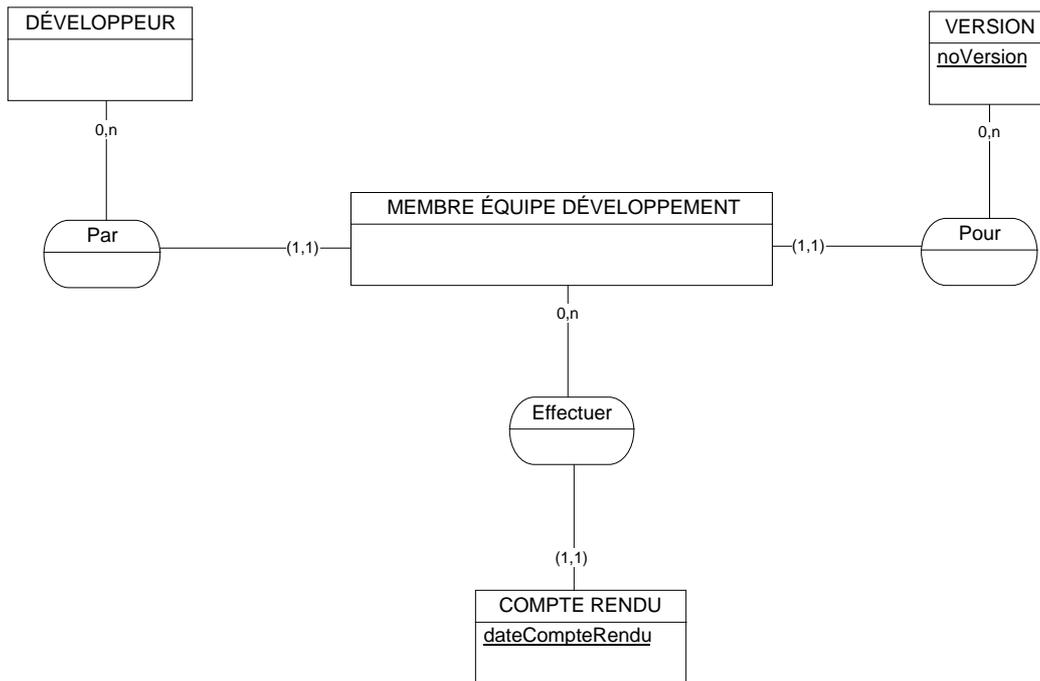
Appliquons ces nouveaux concepts au cas GRC après avoir apporté la précision suivante : une action possède un numéro d'ordre qui l'identifie parmi les actions qui concernent le même tiers. En outre, nous pouvons construire un « agrégat » AGRÉMENT qui évitera la mention de la contrainte d'inclusion.



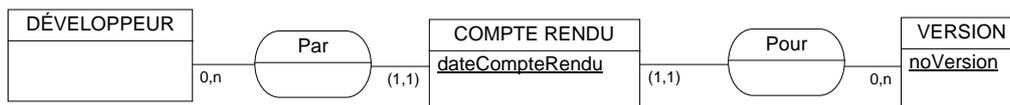
Le lien identifiant porte forcément les cardinalités 1, 1 sur le rôle de l'entité qu'il contribue à identifier. Les parenthèses autour du couple 1, 1 indiquent que le lien participe à l'identification.

Les notions d'identifiant relatif et de lien identifiant permettent de représenter les deux cas typiques évoqués précédemment : l'entité faible (un identifiant relatif + un lien identifiant) et l'agrégat (deux liens identifiants).

Mais elles permettent également de représenter des cas plus sophistiqués. En voici un exemple inspiré du cas Secolog (BTS informatique de gestion, option Développeur d'applications, métropole 2001).



Les développeurs travaillent sur des versions (de logiciels). Chacun fait un compte rendu de sa journée de travail pour chaque version concernée. Le schéma précédent est tout à fait transposable dans le modèle anglo-saxon puisqu'on y trouve d'abord un agrégat (membre de l'équipe de développement) sur lequel est construite l'entité « faible » Compte Rendu. D'un point de vue sémantique, on peut trouver intéressant de disposer de la composition de l'équipe de développement avant même que tout compte rendu n'ait été rédigé. Mais le concept d'équipe est une extrapolation par rapport au sujet qui n'y fait pas allusion. En ne tenant compte que des règles fournies par le sujet, le schéma pourrait être le suivant :



Vous voyez cette fois que, malgré son indéniable « ergonomie », le modèle anglo-saxon ne peut représenter de façon lisible cette situation « hybride », mi-agrégat, mi-entité faible.

**Remarque pour le lecteur familiarisé avec la notion de DF**

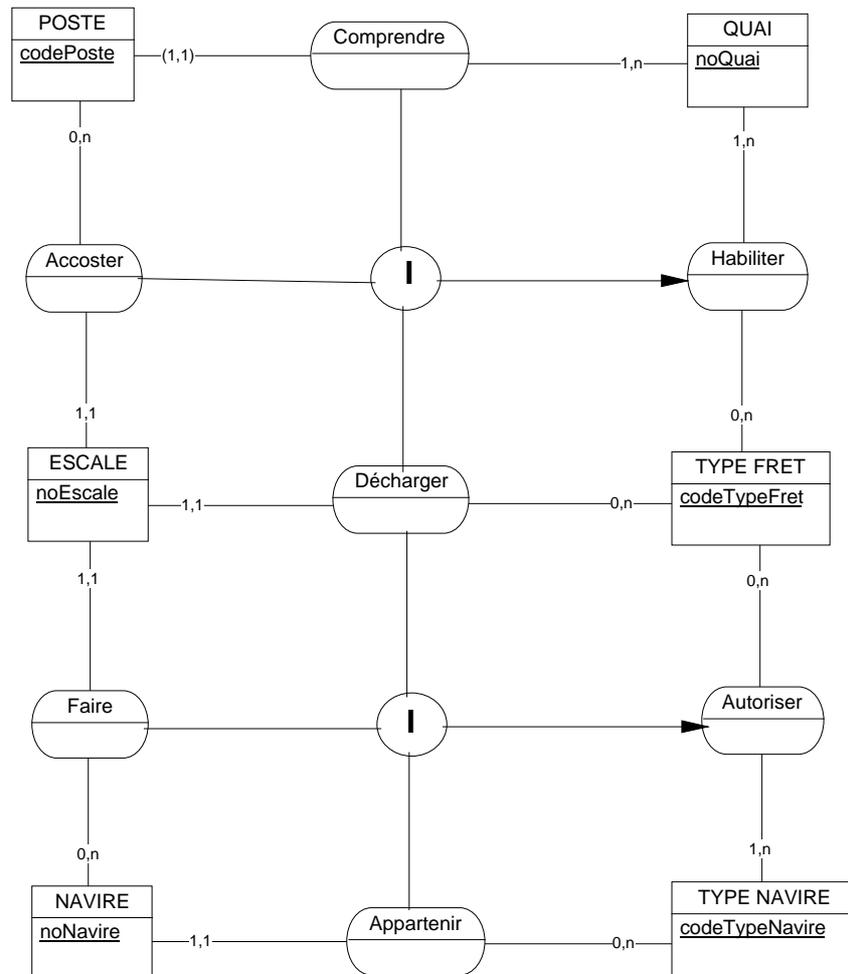
Lorsque nous ne disposons que d'identifiants absolus, il était commode sur le plan pédagogique d'assimiler association fonctionnelle (par exemple, l'association « Réaliser ») et « dépendance fonctionnelle » entre identifiants (noAction → noTiers). Évidemment, la notion d'identifiant relatif introduit une rupture : nous ne pouvons pas écrire « noOrdreAction → noTiers ». Mais, contrairement à ce qui est écrit dans certains documents diffusés lors de la rénovation du BTS CGO (processus 10), cela ne change rien à la nature fonctionnelle de l'association « Réaliser ». Dans sa représentation littérale, elle s'écrit dans les deux cas : **Réaliser(ACTION → TIERS)**. Dire « qu'un couple de cardinalités 1,1 mis entre parenthèses ne correspond pas à une DF » est donc faux, à une DF entre identifiants, certes, mais, si l'on admet la formule, la « DF » entre les entités demeure !

### III.3 - Contre-exemples

L'emploi des « agrégats » et des « entités faibles » est délicat. Voici quelques problèmes liés à leur utilisation :

#### Contre-exemple n° 4 : Lien identifiant et contrainte d'inclusion (Caen-Quistreham)

L'étude de cas du BTS DA (Nouméa 2001) évoquée dans le contre-exemple n° 3 comportait en fait deux contraintes d'inclusion.



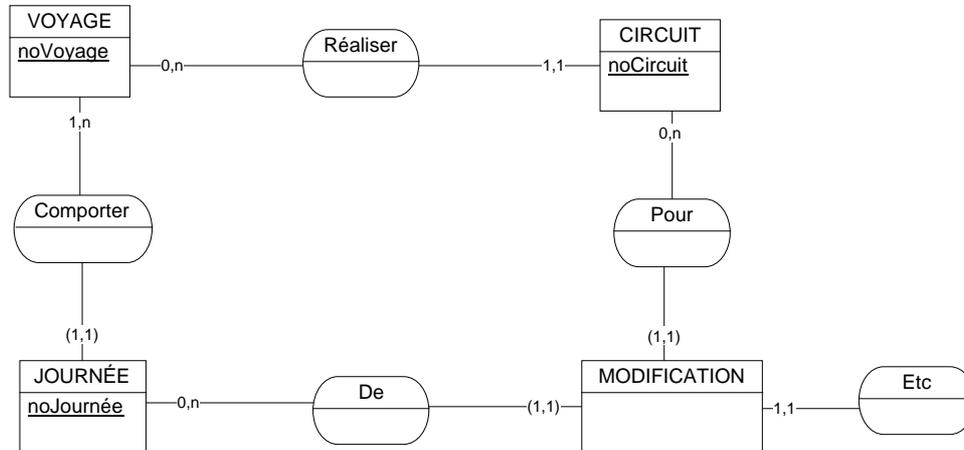
Les deux contraintes reposent sur le même modèle. Dans la première, il s'agit de contrôler que le type de fret déchargé lors d'une escale en accostant un poste compris dans un quai fait partie des types de fret pour lesquels ce quai est habilité. A priori, aucun problème ! Pas d'accord, ont avancé un grand nombre de professeurs car, dans cette contrainte, il y a une « entité faible » et ils ne voient pas la nécessité de mobiliser l'association Comprendre. Nous voyons là une nouvelle conséquence de la confusion entre modèle entité-association et modèle relationnel. Évidemment, dans ce dernier, la relation POSTE a pour clé primaire noQuai, codePoste. En implantant la contrainte d'inclusion à l'aide d'un *trigger*, il ne sera pas nécessaire d'aller plus loin. Mais, c'est oublier que, dans le modèle entité-association, l'identification de l'entité Poste intègre le lien identifiant : l'association Comprendre fait bel et bien partie des sources de la contrainte !

(Décharger ⊗ Accoster ⊗ Comprendre)[QUAI, TYPE FRET] ⊆ Habilitier.

**Contre-exemple n° 5 : Agrégat, l'air !** (Au Fil de la Terre, CAPET IG externe 1997)

Dans ce sujet complexe, il est notamment question de voyages décrits en journée. Un circuit est une « occurrence » de ce voyage à des dates particulières. Pour diverses raisons, il est possible de modifier le contenu d'une journée de circuit.

Le corrigé publié dans le rapport de jury nous propose le schéma suivant (extrait) :



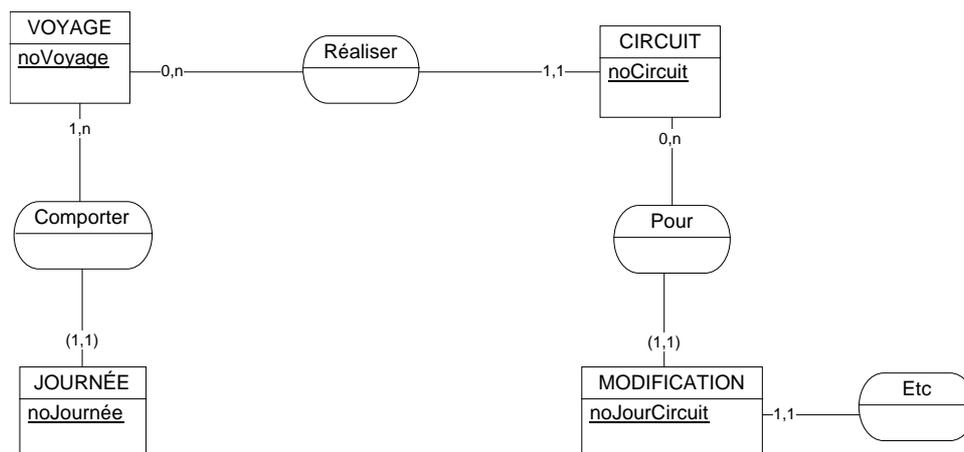
Remarque : L'association Etc a été dessinée pour montrer que la présence d'un agrégat pouvait être justifiée par la nécessité de construire de nouvelles associations.

L'idée est séduisante. Mais, mais... Réalisons le modèle logique relationnel, présenté ci-dessous avec des conventions d'écriture « classiques » :

VOYAGE(noVoyage, ...)  
 JOURNÉE(noVoyage#, noJournée, ...)  
 CIRCUIT(noCircuit, ..., noVoyage#)  
 MODIFICATION(noCircuit#, noVoyage#, noJournée, ...)

Or, dans cette dernière relation, la DF noCircuit → noVoyage, à l'intérieur de la clé, n'est pas tolérable !

La difficulté peut être contournée ainsi :



**Contre-exemple n° 6 : Le lait tourné !** (Dossier informatique, CAPET B externe 2001)

Extrait du sujet :

La production de fromages engendre une organisation particulière quant aux tournées de ramassage de lait, matière première indispensable aux différentes productions de l'entreprise.

La mise en place d'une base de données, permettant de gérer les différentes ressources, a posé certains problèmes compte tenu des contraintes spécifiques aux tournées.

**Travail à faire**

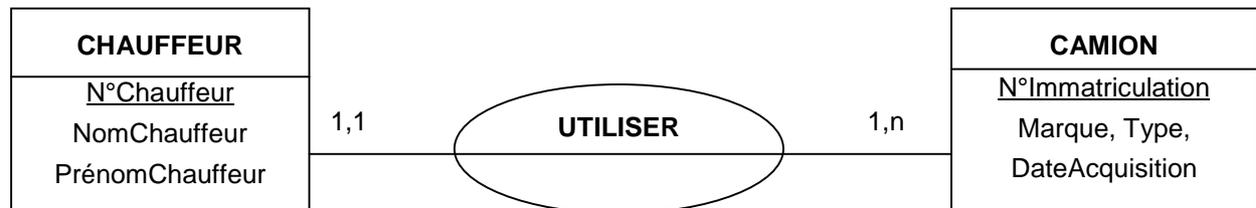
Le responsable vous demande de confirmer ses choix en analysant les différentes solutions proposées et les options retenues.

**4.1** - Pour quelles raisons concrètes les deux représentations de l'annexe 14 n'ont-elles pas été retenues dans le schéma de la base de données de l'annexe 16 ?

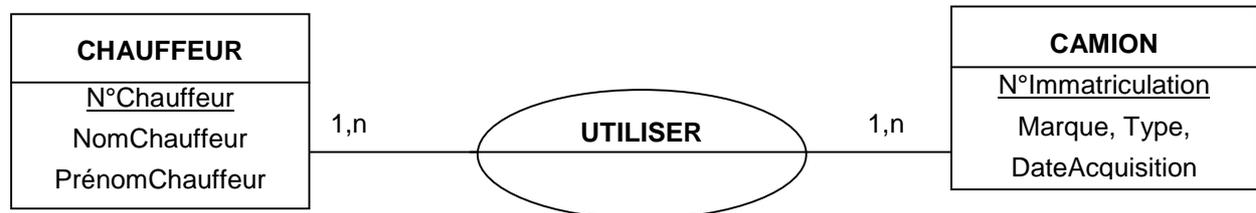
**4.2** - Écrire les **règles de gestion** qui traduisent les deux nouvelles solutions proposées en annexe 15.

**ANNEXE 14 : Extraits de schémas conceptuels des données (Modèles n° 1 et n° 2)**

Modèle 1

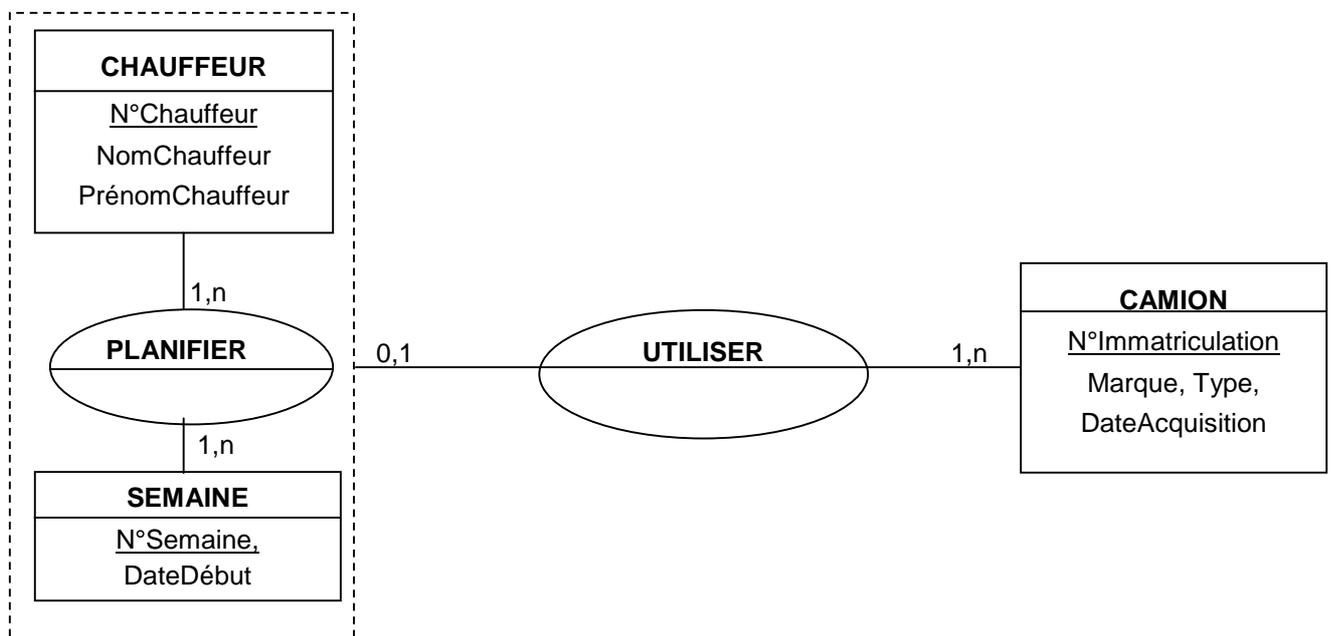


Modèle 2

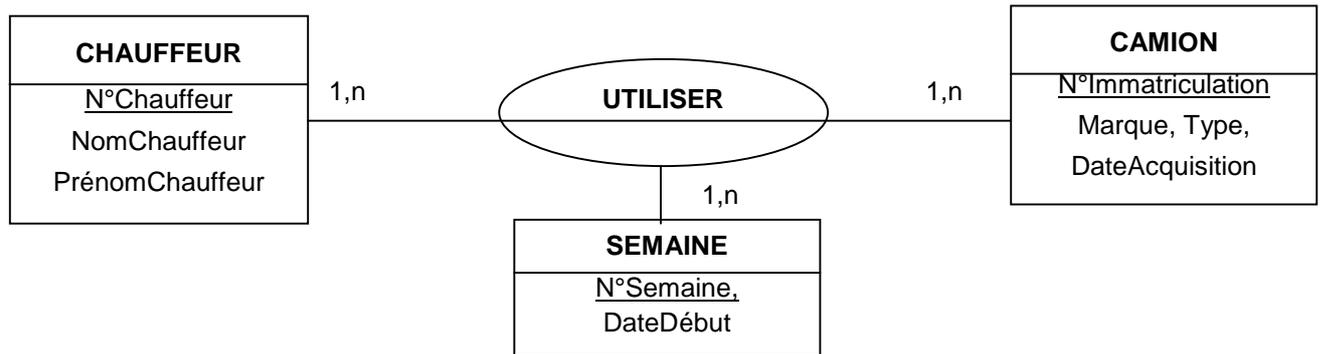


**ANNEXE 15 : Extraits de schémas conceptuels des données (Modèles n° 3 et n° 4)**

Modèle 3



Modèle 4



**ANNEXE 16 : Schéma relationnel complet**

CHAUFFEUR (N°Chauffeur, NomChauffeur, PrénomChauffeur, DateNaissance, Photo)

CAMION (N°Immatriculation, Marque, Type, DateAcquisition, VolumeUtile)

SEMAINE (N°Semaine, DateDébut)

TOURNEE (N°Tournée, NomTournée, NbKms)

PLANNING (N°Semaine, N°Tournée, #N°Chauffeur, #N°Immatriculation)

PRODUCTEUR (N°Producteur, NomProducteur, Lieu-dit, CP, Commune, Quota, #N°Tournée)

LIVRAISON (#N°Producteur, #N°Semaine, Quantité)

Commentaires :

Dans la première question, on veut évidemment amener le candidat à considérer que, dans le modèle n° 1, la « DF » (association fonctionnelle) entre CHAUFFEUR et CAMION est contraire aux règles de gestion. Un chauffeur utilise au moins un camion. Au demeurant, puisqu'un camion est utilisé par au moins un chauffeur, on ne sait pas qui utilise tel camion à tel moment. Dans le modèle n° 1, il s'agit de percevoir que l'unicité du couple CHAUFFEUR-CAMION induite par l'association maillée est également contraire aux règles de gestion : sans doute en considération du temps, un chauffeur peut utiliser plusieurs fois le même camion. Au demeurant, on ne sait toujours pas qui utilise tel camion à tel moment.

Dans la deuxième question, le modèle n° 4 nous offre un nouvel exemple de « fausse ternaire ». Certes, on ne sait pas quelle est l'unité de temps à prendre en compte et notre sentence est peut-être un peu sévère. Il n'en demeure pas moins que, plusieurs chauffeurs pouvant utiliser le même camion la même semaine, on ne sait toujours pas qui utilise tel camion à tel moment.

Le modèle n° 3 utilise un agrégat « à l'américaine ». Sémantiquement, il exprime une petite différence avec le modèle n° 4. Dans ce dernier, l'association ternaire indique que le triplet CHAUFFEUR-CAMION-SEMAINE est créé en une seule opération. Dans le précédent schéma, les cardinalités 0,1 sur le rôle de l'agrégat nous indiquent que le planning de travail des chauffeurs est élaboré prioritairement. L'attribution du camion vient après. Mais, mais... Puisqu'un camion peut être affecté à plusieurs couples CHAUFFEUR-SEMAINE, il peut l'être à des couples pour lesquels seul le chauffeur est différent. Donc, plusieurs chauffeurs peuvent utiliser le même camion la même semaine et on ne sait toujours pas qui utilise tel camion à tel moment. **L'intervention de l'agrégat ne résout donc rien !**

Vous remarquerez que le schéma de la base de données (annexe 16) fait intervenir la notion de tournée sans résoudre ce problème !

## Mets ta modèle et suis-moi<sup>45</sup> !

### IV - Le méta-modèle<sup>46</sup>

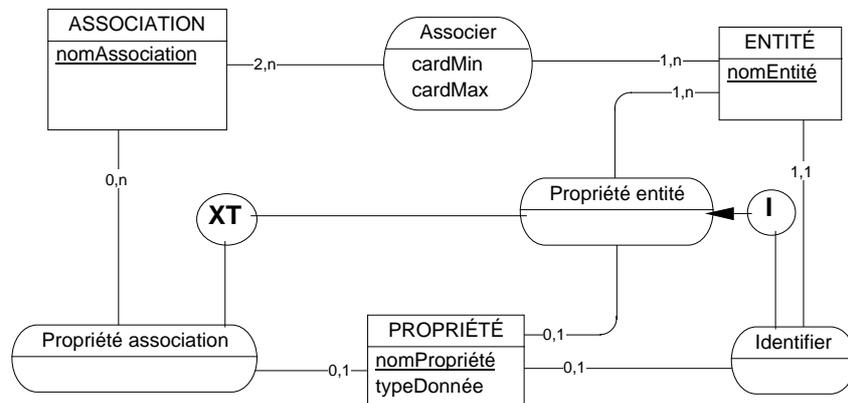
Il nous semble que la meilleure façon de partager les concepts est de les modéliser car cela contraint à les formaliser de façon rigoureuse ! Essayons donc de modéliser les concepts du modèle entité-association à l'aide de ses propres concepts.

#### IV.1 - Les concepts de base

- En première approche, nous concevons le schéma suivant :

Nous faisons les hypothèses suivantes :

- Une association ne possède jamais de propriété identifiante.
- Toute propriété est obligatoire et monovaluée.
- La notion de paramètre n'est pas représentée dans le MCD.



La contrainte de partition exprime qu'une propriété décrit soit une entité soit une association, pas les deux mais forcément une des deux. *Propriété association* et *Propriété entité* n'ont qu'une entité commune dans leur collection, l'entité *PROPRIÉTÉ* : le pivot portant sur cette entité peut rester implicite.

La contrainte d'inclusion exprime que la propriété qui identifie une entité fait partie des propriétés de cette entité. *Propriété entité* et *Identifier* ont la même collection : le pivot portant sur cette collection, le couple *ENTITÉ-PROPRIÉTÉ*, peut rester implicite.

**On ne peut pas valider l'association Associer telle qu'elle est conçue.** En effet, ici, le couple ENTITÉ-ASSOCIATION est forcément unique, ce qui n'est pas toujours le cas dans la réalité (cas des associations réflexives où une même entité joue plusieurs rôles).

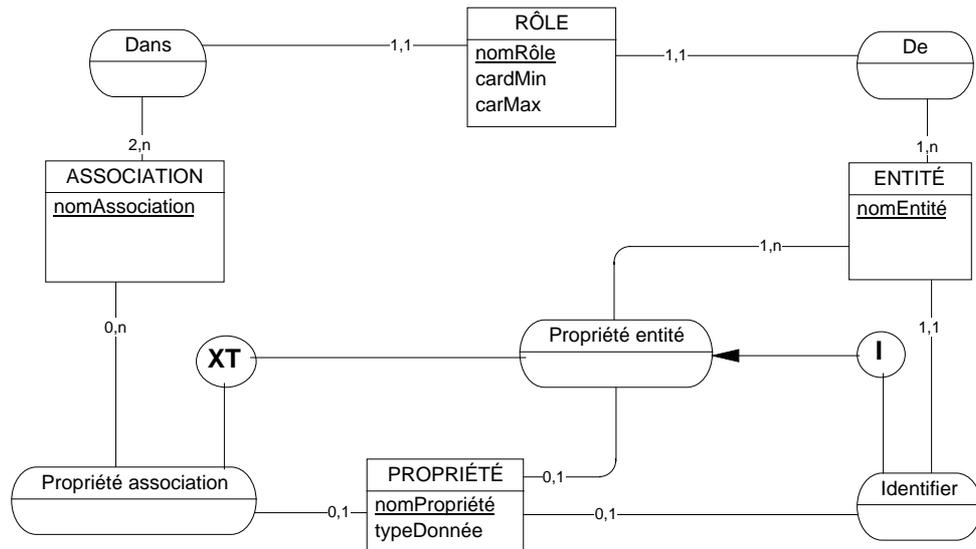
- L'intervention du concept de rôle nous donne :

Nous faisons l'hypothèse suivante :

- Le nom d'un rôle est unique.

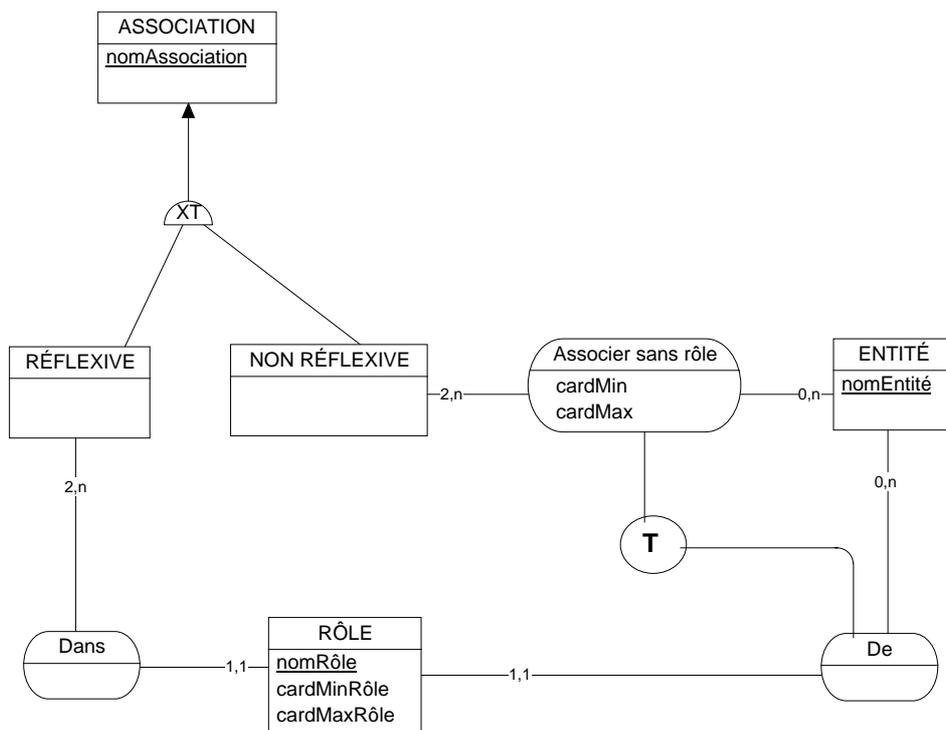
<sup>45</sup> Mille excuses mais mon fils y tenait ☺ !

<sup>46</sup> Cette section est vraiment réservée aux initiés !



**Cette solution ne correspond pas à la pratique** : ici, l'intervention des rôles est systématique alors que, dans la réalité, les rôles ne sont obligatoires que dans les associations réflexives. Pouvons-nous modéliser cela ?

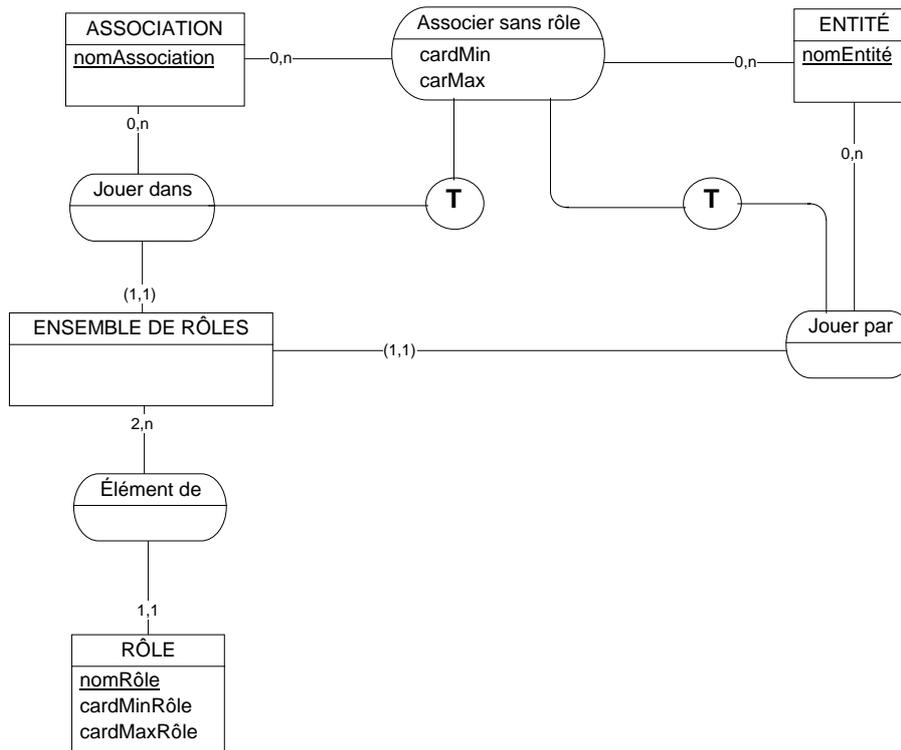
- La prise en compte d'associations spécialisées pourrait nous donner :



La contrainte de totalité signifie que l'entité *ENTITÉ* participe au moins une fois à l'une des deux associations. *ENTITÉ* est la seule entité commune aux deux associations *Associer sans rôle* et *De* : le pivot peut donc rester implicite.

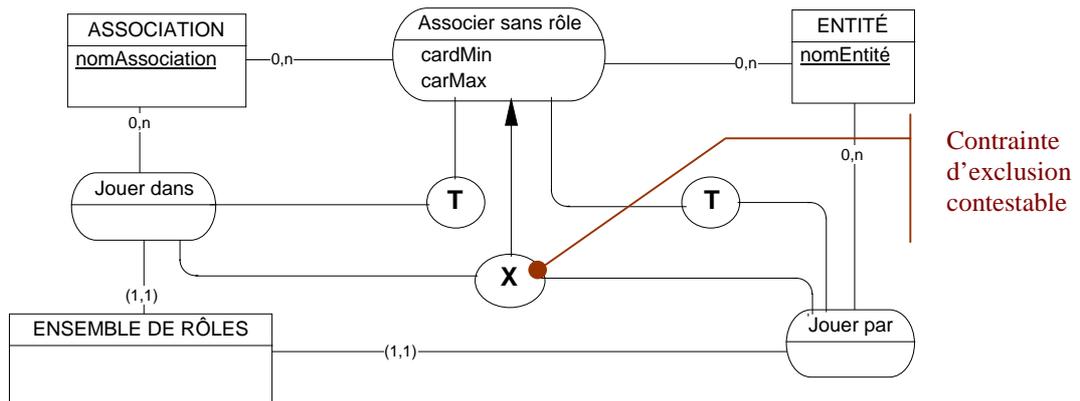
**Ce qui est modélisé ici ne correspond toujours pas à la pratique.** Par exemple, dans une association ternaire où une même entité joue deux des trois rôles, seuls ces deux rôles sont à distinguer par un nom. Par ailleurs, il n'est pas précisé qu'une association réflexive comporte au moins deux rôles portant sur la même entité. Pouvons-nous modéliser cela ?

- L'intervention du concept d'ensemble de rôles joués par une même entité dans une même association nous donne :



ENSEMBLE DE RÔLES est un agrégat. Les pivots des contraintes de totalité (ASSOCIATION d'une part, ENTITÉ d'autre part) peuvent rester implicites.

Nous avons beaucoup progressé. Ici, un ensemble de rôles joués par une entité dans une association comporte au moins deux rôles. Une association peut à la fois réunir des entités distinctes et un ensemble de rôles joués par une autre entité. Le seul point délicat, c'est que cette entité peut être aussi l'une de celles qui sont associées sans rôle. Nous avons déjà vu certains professeurs exprimer ce genre de contrainte d'exclusion de la manière suivante :



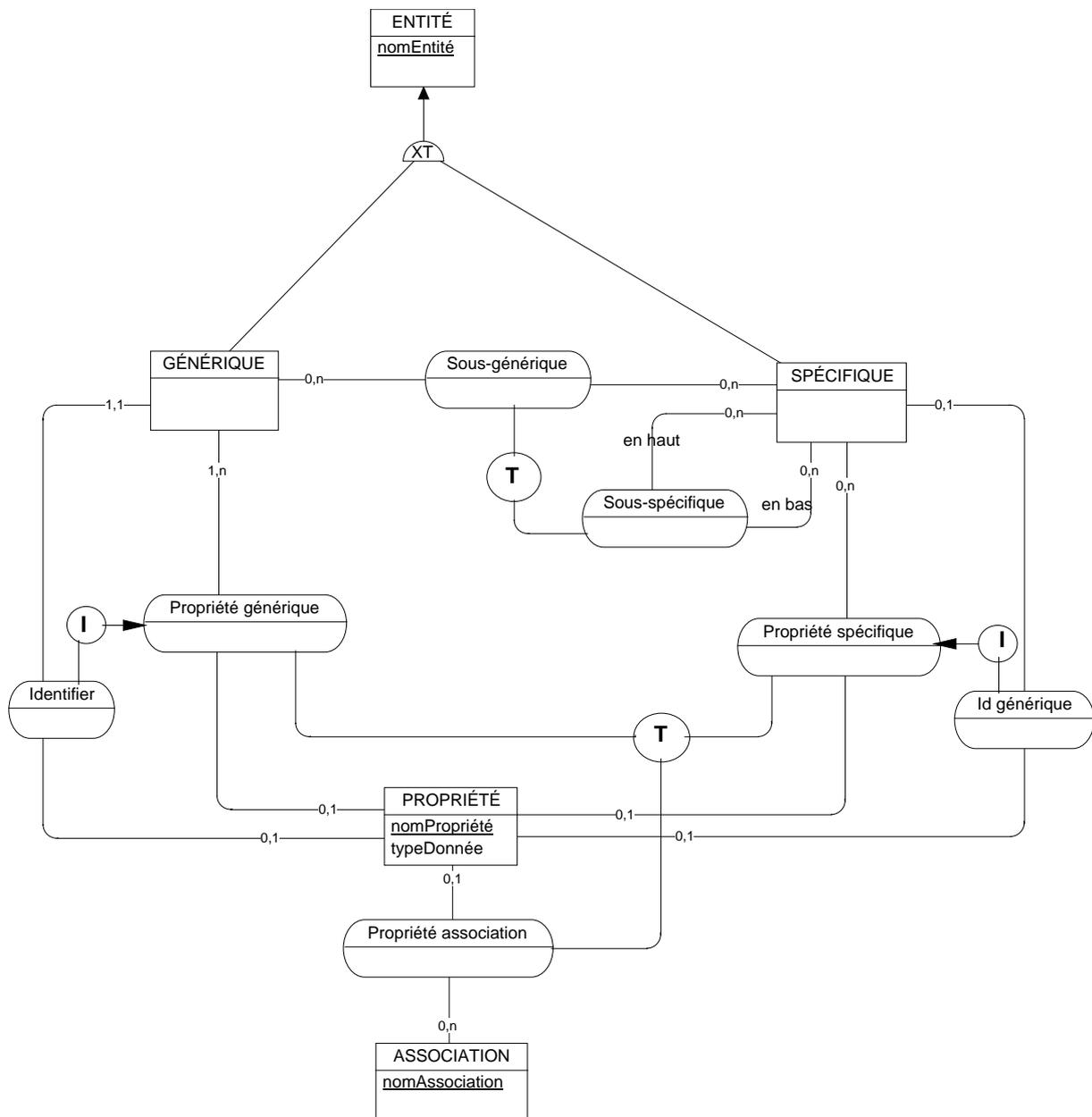
Nous ne connaissons aucun ouvrage documentant cet usage particulier de la contrainte d'exclusion et il nous semble pour le moins incongru qu'une contrainte d'exclusion soit orientée. Nous préférons l'écriture d'une contrainte textuelle qui peut prendre un caractère plus formel :

$$(\text{Jouer dans} \times \text{Jouer par})[\text{ASSOCIATION}, \text{ENTITÉ}] \cap \text{Associer sans rôle}[\text{ASSOCIATION}, \text{ENTITÉ}] = \emptyset$$

## IV.2 - La spécialisation

- Nous partons des hypothèses suivantes :
  - Une entité générique est définie comme une entité n'ayant pas d'entité spécifique ou n'étant pas une entité spécifique.
  - Une entité spécifique spécialise au moins une entité générique ou au moins une entité spécifique.
  - Nous admettons l'héritage multiple. Cependant, il va de soi qu'une entité spécifique ne peut pas spécialiser elle-même.
  - Une entité générique est toujours identifiable.
  - Une entité spécifique peut posséder son propre identifiant.
  - Nous admettons qu'une entité spécifique ne possède pas de propriété.

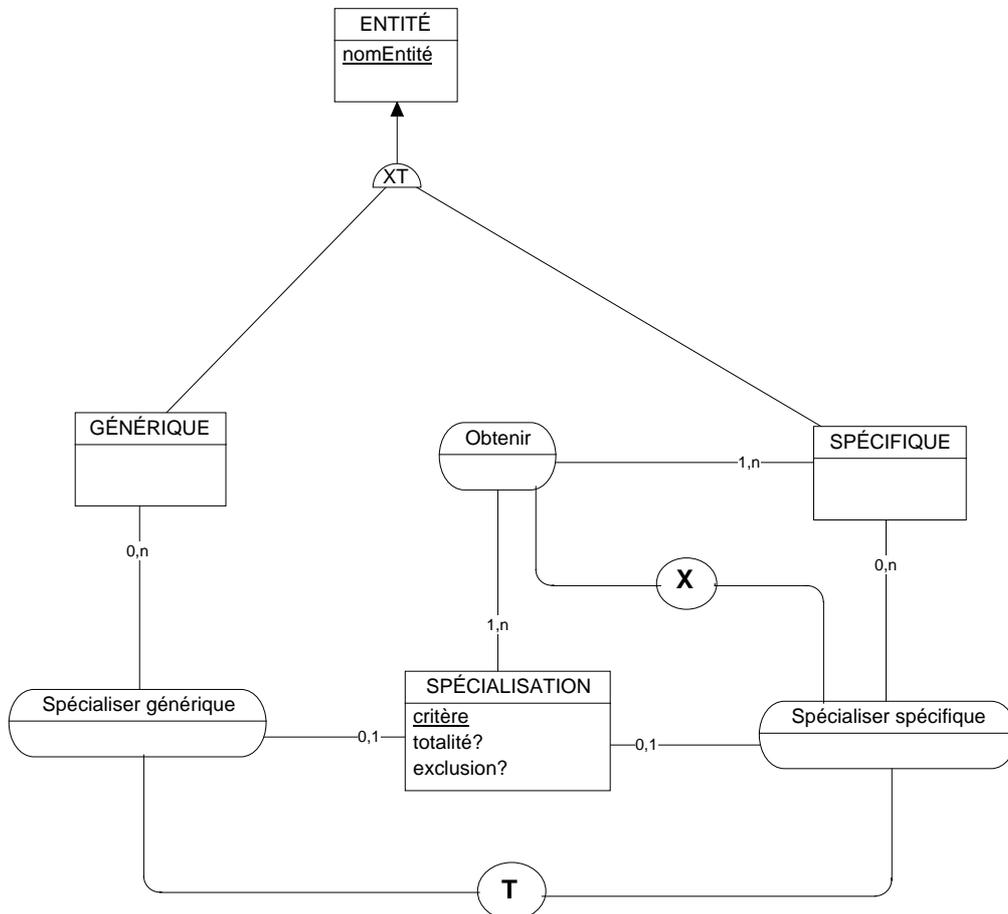
Voici une interprétation possible de ce thème :



Contrainte d'irréflexivité :  $\forall S \in \text{Sous-spécifique} : S.SPÉCIFIQUE_{\text{en haut}} \neq S.SPÉCIFIQUE_{\text{en bas}}$

Cette interprétation nous semble conforme à certaines versions du modèle entité-association étendu mais **pas à celle que nous utilisons pour faire les schémas**. En effet, la demi-lune de Power\*AMC (ou le triangle de WinDesign) représente une spécialisation possible, parmi d'autres à partir d'une même entité comme l'on peut définir plusieurs typologies au sein d'un même concept en considérant différents critères (exemple : les réseaux informatiques).

- En introduisant la notion de spécialisation, nous obtenons :



Nous avons fait l'hypothèse qu'un critère identifie une spécialisation.

Si la notion de spécialisation est prise en compte, on peut la caractériser par la présence ou non d'une contrainte de totalité et la présence ou non d'une contrainte d'exclusion : c'est pourquoi deux propriétés de type booléen ont été placées dans cette entité.

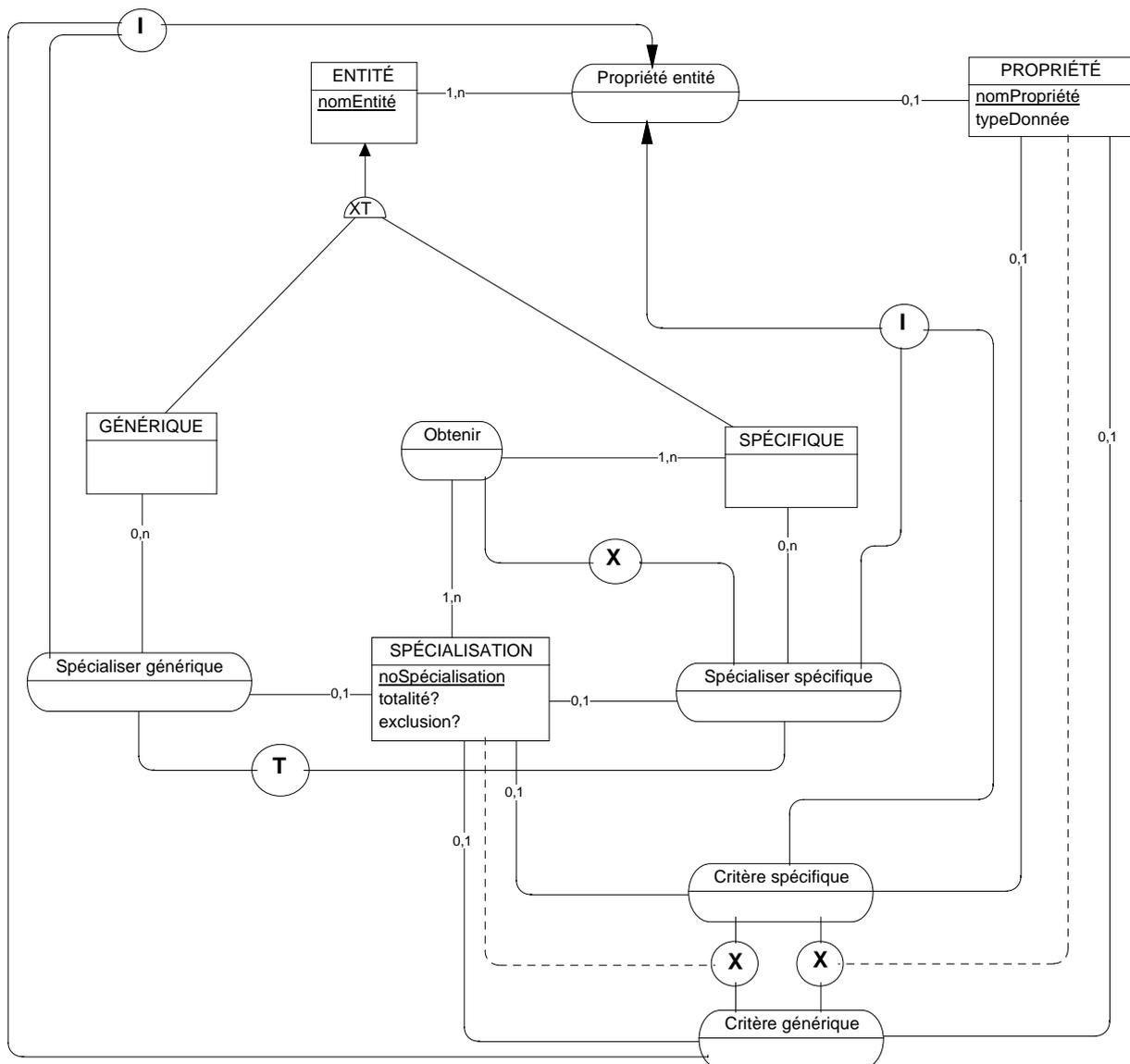
Rappelons que, parmi les entités génériques, nous avons placé celles qui n'ont aucun spécifique : cela justifie la cardinalité minimale 0 sur le rôle de *GÉNÉRIQUE* dans *Spécialiser générique*.

La spécialisation concerne soit une entité générique soit une entité déjà spécifique, pas les deux mais au moins l'une des deux, d'où la contrainte de totalité.

Une entité spécifique est forcément obtenue par au moins une spécialisation et une spécialisation crée au moins une entité spécifique. La contrainte d'irréflexivité devient une contrainte d'exclusion que l'on peut mentionner sur le schéma : dans une spécialisation donnée, l'entité spécialisée ne fait pas partie des entités spécifiques obtenues.

**L'hypothèse émise ne correspond pas à la pratique.** Le plus souvent, au contraire, aucun critère de spécialisation n'est mentionné et, quand il est mentionné, il porte généralement sur une propriété de l'entité spécialisée. Pouvons-nous modéliser cela ?

- En introduisant la notion de critère lié à une propriété, nous pourrions obtenir :

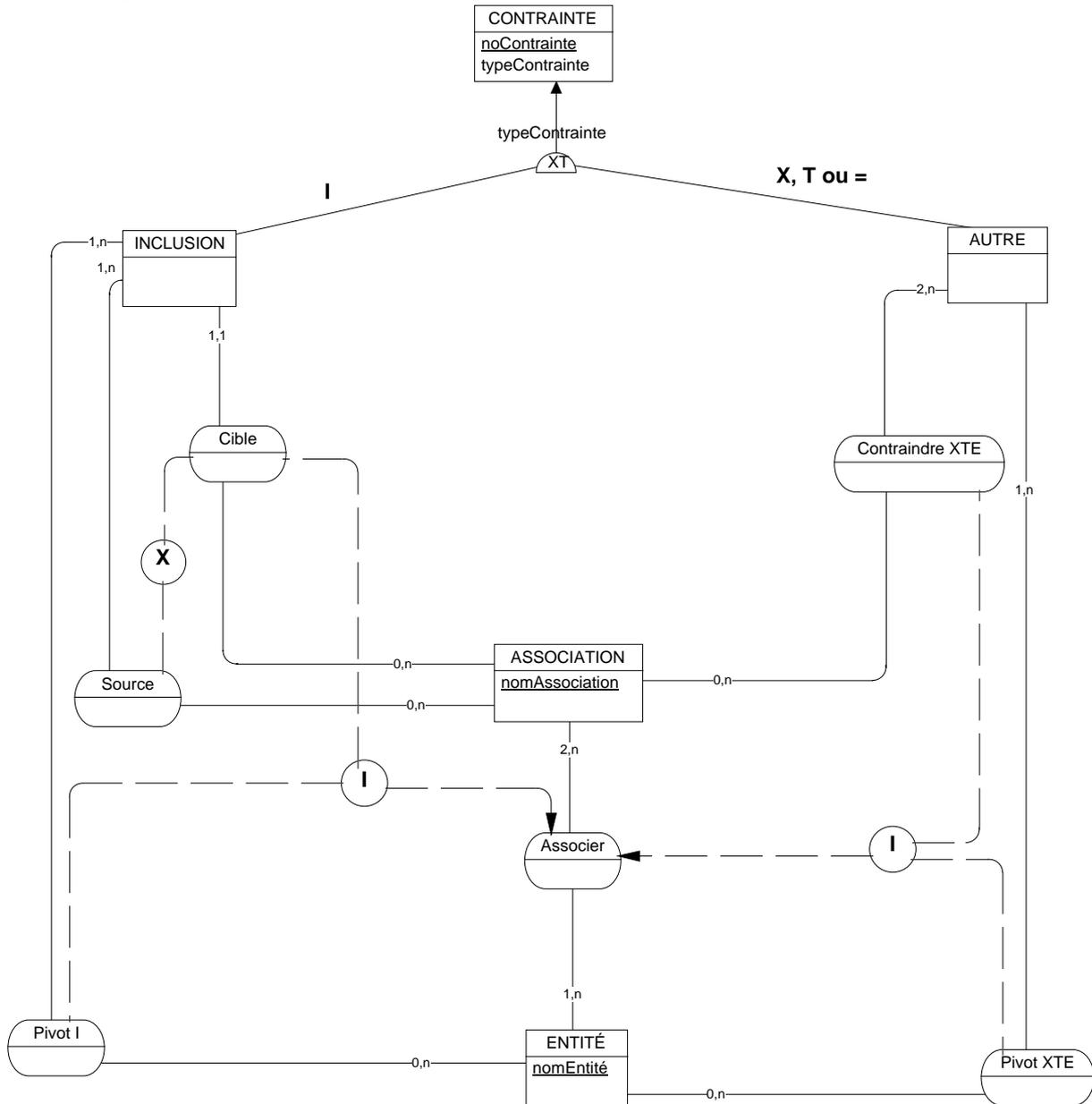


On voit que la « spécialisation » se propage comme la lumière (ou les ténèbres ?).

Nous ne possédons plus aucun moyen d'identifier une spécialisation : **il a fallu créer un identifiant spécifique (hors pratique usuelle)**. Il a fallu spécifier que la spécialisation porte éventuellement sur rien, éventuellement sur une propriété d'une entité spécifique, éventuellement sur une propriété d'entité générique mais pas sur les deux (d'où la contrainte d'exclusion sur les rôles de SPÉCIALISATION). Une propriété est l'objet d'au plus un critère, soit spécifique, soit générique (d'où la contrainte d'exclusion sur les rôles de PROPRIÉTÉ). Enfin, il faut spécifier dans les deux cas que la propriété qui sert de critère fait partie des propriétés de l'entité. Nous l'avons mentionné sur le schéma à l'aide de deux contraintes d'inclusion dont l'usage particulier ici n'est pas documenté : en effet, dans les deux cas, l'une des entités concernées par les associations contraintes est une entité spécifique (soit l'entité GÉNÉRIQUE soit l'entité SPÉCIFIQUE) alors que la cible comporte leur entité générique (l'entité ENTITÉ). **Dans l'état actuel de nos connaissances, nous ne savons pas si cette écriture est valide.**

### IV.3 - Les contraintes inter-associations

- Nous partons des hypothèses suivantes :
  - Une contrainte d'exclusion, d'égalité ou de totalité lie au moins deux associations.
  - Une contrainte d'inclusion possède une association cible et au moins une association source.



Pour améliorer la lisibilité du schéma, les « pattes » des contraintes ont été dessinées en pointillés et **la notion de rôle entre les entités ENTITÉ et ASSOCIATION a été écartée.**

La contrainte d'exclusion signifie qu'une même association ne peut être à la fois cible et source d'une même contrainte d'inclusion.

Les contraintes d'inclusion signifient :

- D'une part, le pivot d'une contrainte d'inclusion fait partie de la collection de l'association cible de cette contrainte.

- D'autre part, le pivot d'une autre contrainte (égalité, totalité ou exclusion) fait partie de la collection de chaque association contrainte.

La contrainte entre le pivot d'une contrainte d'inclusion et les collections des associations sources ne peut être représentée sur le schéma. On peut la formuler ainsi :

$\forall I \in \text{INSERTION} : (\text{Pivot } I[\text{INCLUSION} = 1])[\text{ENTITÉ}] \subseteq (\text{Source}[\text{INCLUSION} = 1] \otimes \text{Associer})[\text{ENTITÉ}]$

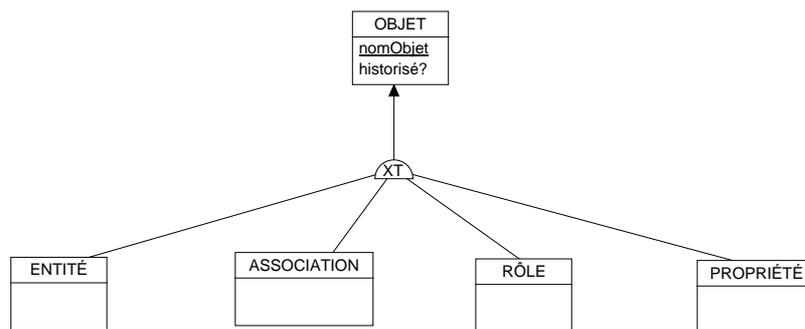
**Écarter la notion de rôle a pour conséquence de spécifier de façon incorrecte les associations *Pivot XTE* et *Pivot I*** : dans une association réflexive source d'une contrainte d'inclusion ou tout simplement frappée par une autre contrainte, il est logiquement possible de trouver dans le pivot plus d'une fois la même entité (dans des rôles différents). Le schéma obtenu serait donc à réviser pour le rendre compatible avec le schéma obtenu en IV.1.

#### IV.4 - Compléments

Dans cette version du document, certaines notions du modèle entité-association étendu n'ont pas été abordées : les spécificités de certains types de données, les propriétés multivaluées, les propriétés composées, les domaines (ou types construits), la généralisation, la stabilité, l'historisation, la spécialisation des associations notamment<sup>47</sup>.

Certaines notions peuvent être aisément intégrées au méta-modèle :

- **La stabilité** s'exprime différemment selon qu'il s'agit d'une propriété (**S**) ou d'un rôle (**V** : verrouillé ; **D** : définitif). Dans le premier cas, *Stable?* devient une propriété de type booléen de l'entité *PROPRIÉTÉ*. Dans le deuxième cas, il y a une logique ternaire puisqu'une « patte verrouillée » est forcément « définitive ». On peut opter pour des propriétés de type énuméré : *stabilité* dans l'association *Associer sans rôle* et *stabilitéRôle* dans l'entité *RÔLE*.
- **L'historisation** s'exprime de la même façon (**H**) quel que soit « l'objet » concerné : propriété, entité, association ou rôle. Nous pourrions généraliser ces « objets », notre convention d'écriture sur les désignations d'objet nous permettant de disposer à coup sûr d'un identifiant.



**Cela ne suffit pourtant pas** car la notion de rôle n'a pas été systématisée et l'on trouve des « rôles » dans l'association *Associer sans rôle*.

- **La notion de type** peut donner naissance à une entité et **la notion de domaine** s'interposer entre les entités *PROPRIÉTÉ* et *TYPE*, etc.

D'autres notions sont plus délicates à modéliser, la spécialisation des associations notamment. De ce fait, nous ne sommes pas loin de nous poser la question : le modèle entité-association sait-il vraiment parler de lui-même ?

<sup>47</sup> La dépendance fonctionnelle (ou contrainte d'unicité) entre rôles dans une association relève d'une problématique particulière analysée dans « Vraies ternaires, fausses ternaires et agrégats ».

## Conclusion

Dans ce document, nous n'avons pas recherché l'exhaustivité, le choix contraire fut même conscient.

Nous avons voulu d'abord positionner les capacités d'abstraction utilisées dans le modèle de référence (le MEA) par rapport aux capacités qu'utilise tout autre humain doué de raison. Nous en avons volontairement limité la panoplie. Il nous semble cependant que la réflexion didactique sur laquelle repose notre enseignement devrait s'intéresser à d'autres capacités d'abstraction.

- **L'exception**<sup>48</sup>. Prenons l'exemple des oiseaux qui ne volent pas tous. La façon dont nous avons abordé la spécialisation nous conduira à distinguer les oiseaux qui volent et les oiseaux qui ne volent pas, deux catégories complètement disproportionnées. Le problème se complexifie si l'on considère que la classe des oiseaux hérite de celle des animaux volants... Mentalement, vous savez bien qu'il est préférable de faire de l'aptitude au vol une caractéristique générique des oiseaux, propre ou héritée, et d'indiquer les exceptions (le fameux lien « sauf » dont les enfants usent dès l'école primaire pour apprendre les règles de grammaire). Mais les méthodes de conception par les objets elles-mêmes intègrent encore mal cette capacité<sup>49</sup>.
- **La synthèse**. Sur le plan didactique, le professeur qui enseigne la modélisation ne peut l'éviter sans... provoquer des dégâts collatéraux. Ainsi, à quoi sert-il de convaincre un élève que l'affirmation « les clients commandent des articles » est fautive, qu'il faut dire « une commande est passée par un client et une commande porte sur des articles » ? A quoi ça sert, et surtout qu'est-ce que cela dessert d'apprendre incidemment à un élève qu'une synthèse est une erreur ? Dans nombre de ses diagrammes, UML propose différents niveaux de granularité, ce qui conduit à prendre en compte l'activité de synthèse.
- **L'analogie**. L'acte pédagogique, quel qu'il soit, recourt souvent à l'analogie. L'enseignant l'utilise quotidiennement mais il cherche aussi à développer la capacité de l'élève à l'exploiter. L'analogie de structure se rapproche de l'activité de généralisation, l'enseignant la privilégie. À l'opposé, la tendance « naturelle » d'un grand nombre d'élèves est de s'intéresser aux analogies de surface, en quelque sorte aux étiquettes plutôt qu'au contenu et à son organisation. Cela n'est évidemment pas sans conséquence dans une activité de modélisation de données ou de processus. La programmation par les objets possède un objectif de réutilisation des classes développées, que le programmeur choisira notamment... par analogie !

Nous avons ensuite voulu montrer que l'expression des contraintes transpose le plus souvent, dans le domaine de la modélisation, des concepts mathématiques relativement élémentaires. Il nous semble que le rapport établi entre la représentation graphique du modèle et une écriture littérale plus « formalisante » en facilite la compréhension.

Enfin, nous avons souhaité rendre à César (Oscar ?) ce qui lui appartenait et, ce faisant, entrevu que la qualité d'un mode de représentation ne se résume pas à son ergonomie.

Au passage, nous espérons avoir illustré les vertus cardinales que nous tentons de maintenir dans notre activité d'enseignement : l'interdisciplinarité, la rigueur scientifique, l'esprit critique, l'humour et... l'humilité !

---

<sup>48</sup> Précisons que nous traitons principalement de connaissances déclaratives et non de connaissances procédurales, domaine dans lequel le mot « exception » possède une signification quelque peu différente.

<sup>49</sup> On ne peut en effet assimiler complètement le mécanisme de surcharge (redéfinition) avec le mécanisme d'exception (masquage, rupture du lien d'héritage) lié à ce que les experts appellent la *typicalité*. Il est vrai que l'exception pose des problèmes théoriques délicats, la contradiction par exemple. On voit bien ici ce qui sépare encore un système de connaissances et un système logiciel : beaucoup d'entre nous arrivons à vivre avec nos contradictions, à décider malgré elles... Ces problèmes sont abordés de façon précise notamment dans *Les langages à objet de Gérald Masini et alii chez InterÉditions et L'intelligence artificielle et le langage de Georges Sabah chez Hermès*.