

MODÈLE RELIÉ : Noyau d'amorçage

Richard Chappuis

CH-2900 Porrentruy – ric@act.ch

Résumé

Le procédé présenté dans cet article propose principalement une nouvelle méthode pour modéliser et archiver des informations. Il existe peu de modèles de données capables d'archiver efficacement et simplement un grand nombre d'informations organisées, on pourra donc comprendre ce procédé comme étant une alternative au modèle relationnel [1]. Ce procédé est d'ailleurs proche du concept fondateur du modèle relationnel, puisqu'il est également basé sur des *relations*. Cependant il s'en distingue dans la mesure où il est *exclusivement* constitué par une pluralité de relations, avec pour corollaire, que sa mise en œuvre est totalement différente et unique.

La portée du procédé dépasse largement le seul domaine de l'archivage et de la modélisation d'informations : elle s'inscrit dans le cadre plus global du *traitement de l'information*. Il s'agit d'une approche radicalement novatrice, qui redéfinit la notion d'information et qui permet de *traiter l'information autrement*.

Cet article constitue une introduction au procédé ¹, et s'intéresse plus particulièrement à la problématique d'*amorçage d'un système* de traitement d'informations réalisé à l'aide de ce procédé, et d'un *Noyau* théorique qui permet d'initier un tel système.

Mots-clés

Modèle de données, modèle relationnel, modèle objet, postulat, axiome, langage formel, modélisation, relation, lien, quadruplet, unité d'information, traitement d'informations, méta-informations, méta-données, sémantique, temps.

Présentation des sections

Faute d'un *modèle de données* adéquat, les systèmes actuels de gestion d'informations génèrent plus de complexité qu'ils ne sont capables d'en résoudre (cf. section 1). Il y a un problème de fond.

Le procédé remet en question la notion d'information, il est fondé sur un nouveau postulat (cf. section 2), qui permet de représenter et de traiter la *totalité de l'information* à l'aide d'une seule structure de donnée, appelée le *Lien* (cf. section 3). Par "totalité de l'information" on entend également les méta-informations, les informations de description. Ce procédé utilise son propre vocabulaire (cf. section 4). Il est en partie introduit par un exemple simple (cf. section 5) expliqué à l'aide d'une notation graphique *Contenant-Relation* (cf. sous-section 4.2).

Un système de traitement d'informations basé sur ce procédé est amorcé par un *Noyau* (cf. section 6). Un *Noyau* est capable de décrire et de rendre accessible sa propre information de description. Un *Noyau* est constitué du minimum d'information nécessaire à l'*enrichissement* et à l'amélioration de sa propre définition (cf. section 7). Dans cet article, l'accent est mis sur la construction d'un tel *Noyau*.

En conclusion, certaines perspectives technologiques du procédé sont rapidement esquissées (cf. section 8). Les tables des figures, illustrations et tableaux résumant tous les graphiques présentés dans cet article sont regroupées dans la section 9. laquelle précède les Notes, les Références, les Remerciements et la Tables des matières.

1 Introduction

Le procédé propose un nouveau modèle pour *décrire des agencements* complexes d'informations, *et*, pour *archiver simplement* ces agencements et les informations qu'ils contiennent. Ce modèle apporte une solution aux problèmes de segmentation sémantique et de segmentation temporelle de l'information.

1.1 Historique

Lors de la conception du modèle relationnel [1], à la fin des années 60, des questions fondamentales sur la *notion d'information* ont été posées. L'idée d'*archiver l'information (les liens) qui met en relation des données* a été proposée, mais pour des raisons de contraintes technologiques de l'époque et aussi de fiabilité, elle n'a pas été retenue, ou elle n'a pas été suffisamment examinée... C'est justement ce que le procédé présenté ici propose d'examiner en détail.

Le modèle relationnel a cependant gardé son intitulé "relationnel", il s'est imposé comme standard des bases de données, et il archive des *données* et des *références sur des données*.

1.2 État des lieux

Les langages objets permettent certainement de décrire des informations complexes, mais les problèmes de *persistance* des objets et de leurs agencements ne sont pas résolus de façon satisfaisante. Les données contenues dans un objet sont cachées par une interface et sont accessibles via des méthodes d'accès déclarées par cette interface. Les objets d'une base de données objets [2] ne sont ni physiquement archivés comme les enregistrements de taille fixe d'une table, ni simplement agencés comme des contenus prédéterminés regroupés dans un contenant, mais ils sont uniquement accessibles via des appels imbriqués de méthodes d'accès. Les modèles de données permettant d'archiver les objets ne se formalisent pas facilement, ils sont trop complexes et variés pour être couramment adoptés.

La solution généralement utilisée est la technique qui consiste à stocker les objets dans une base de données relationnelle (Object-relational mapping [3]), mais c'est un pis-aller, accepté à défaut d'autre chose, qui néglige les aspects fondamentaux de la *disparité des deux modèles* (relational-object impedance mismatch [4]). De plus, afin de maintenir de bonnes performances, des compromis doivent être faits pour chacun des modèles : les modèles relationnels doivent rester simples pour que l'accès aux données reste efficace si la montée en charge augmente; les modèles objets doivent rester simples pour que la persistance de graphes d'objets dans des tables ne devienne pas problématique, et pour que les modèles relationnels correspondants restent simples et efficaces.

1.3 Conséquences

La *description* d'agencements complexes d'informations est elle-même décrite à l'aide d'informations. Aujourd'hui, les *méta-informations* (les informations de description) nécessaires à la modélisation d'un domaine ou d'un métier particulier, deviennent rapidement difficiles à gérer. Elles sont *segmentées* et *dupliquées* par les couches des architectures 3-tiers [5] et par l'utilisation conjointe des différentes technologies indispensables à la réalisation de systèmes de gestion d'informations (cf. Illustration 1, p. 4).

La *communication* est faite par des fichiers XML, le *stockage des ressources* passe par une base relation-

nelle, l'implémentation de la *logique métier* est écrite en orienté objet, les interfaces graphiques de *présentation* utilisent des techniques telles que HTML, Flash, JavaScript, Ajax...

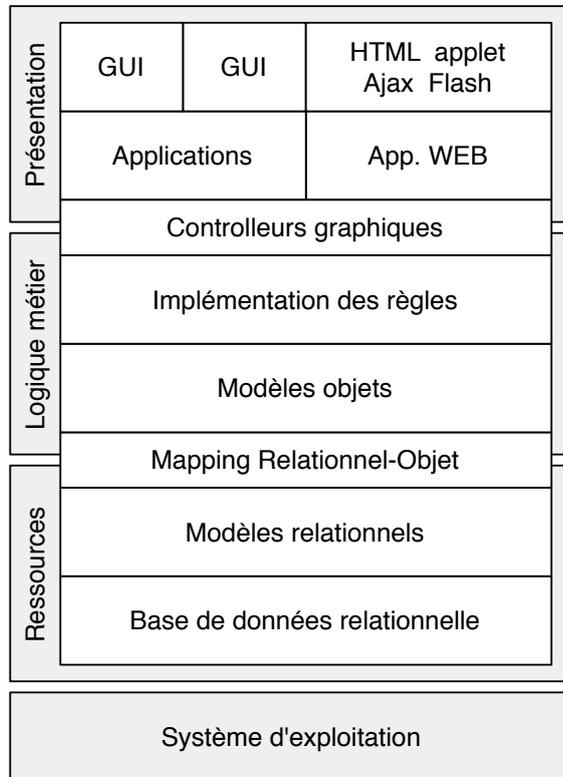


Illustration 1 : les couches d'une architecture 3 tiers

Les *méta-informations* sont **segmentées**, elles se trouvent : dans les modèles relationnels, dans les modèles objets et ses interfaces, dans l'implémentation du code objet, dans les bibliothèques et les frameworks utilisés, dans des fichiers de configuration, parfois dans les couches de présentations graphiques, ou encore dans les arborescences du système de fichiers, dans les configurations du système d'exploitation, etc...

Les *méta-informations* sont **implicites**, elles obéissent à des syntaxes, à des grammaires ou à des règles qui ne sont pas explicitement décrites par de l'information accessible. Elles sont implicites parce que l'organisation de structures de fichiers ou de formats de données est faite à partir de descriptions préconçues, qui sont détaillées quelque-part, dans des documents externes. Par exemple, une hiérarchie préétablie de certains dossiers d'un système de fichiers représente une méta-information implicite.

Les *méta-informations* sont **dupliquées**. Par exemple, dans une architecture 3-tiers, les méta-informations qui décrivent les modèles relationnels (les schémas de la base) sont dupliquées pour décrire leurs correspondances avec les modèles objets (les interfaces, les diagrammes de classes), elles sont dupliquées encore une fois par l'implémentation des méthodes d'accès du code objet. Autre exemple, quand il s'agit de traiter des informations communiquées par un fichier, on les importe dans un schéma de données particulier, ce schéma de données représente une *duplication* des méta-informations décrivant la structure du fichier. Le code responsable de l'importation doit connaître la structure du fichier et celle du schéma de données, les méta-informations sont à nouveau dupliquées.

En conséquences, tout devient très vite *complexe et compliqué* ! De nombreux problèmes surgissent : d'interdépendance, de synchronisation, de gestion de versions, et d'interopérabilité entre les multiples représentations des méta-informations. Une modification apportée aux méta-informations aura des répercussions *mal maîtrisables* sur l'ensemble de l'architecture d'un système.

A défaut de connaître rapidement et précisément les impacts d'une modification sur le système, de peur de créer de nouveaux bogues, la stratégie souvent adoptée est celle de l'*ajout*. Ajout par copier-coller-modifier de classes, de tables, d'attributs, de méthodes..., et par conséquent, des parties importantes de méta-information deviennent orphelines, alourdissant davantage la gestion du système.

La *non-maîtrise des méta-informations* impose une approche guidée par des contraintes techniques plutôt que par les demandes du marché, elle exige une pré analyse complète d'un domaine, elle exige de *figer sa modélisation* en figeant le design de la base de données et des couches inférieures. Autrement dit, la non-maîtrise des méta-informations impose de penser le marché à l'avance, parce qu'il deviendra vite laborieux de faire évoluer la modélisation du domaine traité, pour l'adapter rapidement et répondre favorablement aux demandes ultérieures dudit marché. Il en résulte des systèmes de gestion d'informations *rigides et fragiles*, parce que compliqués et fastidieux à modifier, et donc difficile à maintenir et extrêmement coûteux.

Même dans les cas où les domaines à traiter sont simples, les systèmes de gestion d'informations requièrent une gestion complexe des méta-informations pour garantir leur fonctionnement : la gestion des méta-informations du système ne reflète pas la "simplicité" du domaine traité. Ces systèmes étant justement mal adaptés à la modélisation d'informations complexes, il devient difficile voire impossible d'imaginer un *méta-système* capable de gérer la complexité et l'interdépendance des méta-informations utiles à leur seul fonctionnement. Ces systèmes de gestion d'informations *génèrent plus de complexité* qu'ils ne sont capables d'en résoudre. C'est un problème de fond, et on devine l'impasse dans laquelle se sont engagés les systèmes actuels :

il n'est plus possible d'utiliser l'outil pour l'améliorer lui-même !

1.4 Segmentation sémantique

Ce qui est en cause, c'est le postulat sur lequel se basent les techniques antérieures et qui oblige une *segmentation sémantique* de l'information, isolant les données de ses structures, isolant les structures des formats qui les utilisent, isolant les formats des applications qui les traitent... autrement dit, isolant l'information de sa méta-information.

2 Postulat

Les techniques antérieures reposent sur le postulat que *l'information réside dans les données*. Par conséquent, le *traitement de l'information* signifie le *traitement des données*.

Les techniques antérieures :

- agencent des *données* et des *références sur des données*, elles
- archivent des *données* et des *références sur des données*.

Previous technics	Data	References to Data	
OS	Files	References to Files	Path
R-DB	Records	References to Records	Foreign Key
OO	Objects	References to Objects	Pointer
WWW	Pages	References to Pages	URL
RDF	Resources & Literals	References to Resources & Literals	Triple

Tableau 1 : les techniques antérieures traitent des données et des références sur des données

Mais où se trouve véritablement l'information, quelle est la nature réelle d'une information, qu'est-ce qu'une information ?

2.1 Qu'est-ce qu'une information ?

“Nous partirons du point de vue que rien n'a de sens en soi, mais seulement par rapport à d'autres significations que nous connaissons déjà.” [11]

Partons d'un exemple simple : est-ce que la donnée “richard” est une information ? A priori, dans un contexte culturel précis, la donnée “richard” est porteuse de nombreuses informations : elle peut être reliée à un mot (assemblage de lettres), à un prénom (“Richard” Wagner), à un nom de famille (Pierre “Richard”), à un nom commun français (“richard”, riche), à un personnage célèbre (“Richard” cœur de lion), etc.

Mais en réalité, l'affirmation selon laquelle la donnée “richard” est porteuse de nombreuses informations est biaisée par l'à priori qui spécifie “dans un contexte culturel précis”. En effet, dans un contexte culturel complètement différent, si la donnée “richard” ne peut pas être reliée à un modèle quelconque ou simplement être reliée à une autre donnée, elle peut perdre toute sa signification et n'être porteuse d'aucune information. Peut-être supposera-t-on seulement que “richard” désigne un prénom, ou qu'il correspond à un signe d'une civilisation étrangère, à moins qu'il n'évoque qu'un graffiti sans aucun sens particulier. L'information transportée par la donnée “richard” n'a donc pas grand chose à voir avec la donnée elle-même, mais bien avec *les liens* que cette donnée peut tisser avec d'autres données.

Dans la même logique, si, d'un contexte culturel complètement différent, on isole maintenant la donnée "richard" de tout contexte, si on la place dans un absolu fictif, sans haut ni bas, sans passé ni futur, sans aucune référence à quelque chose d'extérieur, elle n'a plus aucun sens, elle n'est porteuse d'aucune information : elle perd son existence.

Les liens donnent corps à la donnée, ils forment le contexte particulier qui lui donne du sens, qui lui assigne une information propre. Mais si une donnée n'est porteuse d'informations que dans un *contexte précis*, elle est également porteuse d'informations que pour un *temps précis*. La notion de temps est de première importance pour définir *ce qu'est* l'information. ***L'information s'exprime dans le mouvement, elle se comprend par le geste.*** L'information pertinente se trouve dans la transformation et dans le *déroulement* de l'information dans le temps. Autrement dit, une donnée est autant porteuse d'informations par *l'évolution de ses liens dans le temps*, que par ses liens.

2.2 Un nouveau paradigme

Il s'agit d'un nouveau paradigme parce que le procédé est *fondé sur un axiome* puissant et qu'il représente un nouveau modèle de penser *réorientant complètement le traitement de l'information*.

L'axiome s'énonce ainsi : une donnée n'a pas d'existence dans l'absolu, elle n'existe que *par rapport* à d'autres données [7], elle n'existe que par les liens qu'elle entretient avec d'autres données, dans un *temps précis*.

Mais ces "autres données" existent elles aussi par rapport à d'autres données auxquelles elles sont *reliées*... et ainsi de suite, de façon récursive. Finalement il n'y a *plus de données*, mais que des liens qui relient des liens. Tout est relié. D'où le ***nouveau postulat*** qui est à la base des réflexions qui soutiennent le procédé :

L'information réside dans les liens et dans l'évolution de ces liens dans le temps

Toute chose *est* par ses liens avec les autres choses qu'elle *n'est pas*, et il en va de même pour ces autres choses. Une chose se définit par ce qu'elle n'est pas. Ce qui est, est par ce qu'il n'est pas. Un contenant existe par son contenu, car si la notion de contenu n'existait pas, la notion de contenant n'existerait pas non plus. Mais comment amorcer l'existence d'une première chose à défaut d'une autre chose ? A l'instant de la première chose, il y a simultanément (1) cette chose, (2) ce que cette chose n'est pas, et (3) l'espace qui les relie : le lien. Cette première chose (le Contenant), ce qu'elle n'est pas (le Contenu), et le lien qui les relie, ces trois éléments initiaux s'expriment selon le procédé par une seule et même chose, le premier Lien (cf. section 6. Construction d'un Noyau et sous-section 6.4).

L'aspect "temporel" du procédé n'est pas développé ici (cf. sous-section 8.1). Seule la première partie du postulat, "*l'information réside dans les liens*", est présentée. Elle suffit pour évoquer le propos de cet article qui se concentre sur la formation d'un noyau d'amorçage d'un système de traitement d'informations basé sur des liens. D'ailleurs, "l'amorçage d'un système" signifie que toutes les conditions nécessaires au lancement initial d'un système sont présentes à l'instant de ce lancement, ainsi la dimension temps n'intervient pas, elle ne jouera un rôle qu'à partir de cet instant là.

3 Le Lien

Le procédé ne traite pas des données, mais des liens. Et chacun de ces liens est câblé avec d'autres liens, dans un temps précis. Par conséquent, le *traitement de l'information* signifie le *traitement de liens*; le procédé *agence des liens*, il *archive des liens*.

Le “Lien” du procédé, écrit avec une majuscule, est une structure de donnée particulière qu’il ne faut pas confondre avec une simple référence ou un pointeur, ni avec un hyperlien. Les Liens *relient* des Liens, ils contiennent une information qui va au-delà d’une simple référence.

Il ne faut pas confondre non plus, le Lien du procédé avec les “associations” du “modèle associatif” [8], un autre modèle de donnée qui utilise également des “relations”, mais qui s’appuie sur deux structures différentes, les “relations” et les “entités”, et dont sa formalisation n'a aucune correspondance avec le procédé présenté ici.

3.1 Structure du Lien

La structure minimale d’un Lien est composée d’au moins 4 champs : un identifiant unique (id), et trois références (P, R, E). La figure 1a illustre un Lien. Pour la démonstration du procédé², on dira que chacune des références (P, R, E) du Lien indique l’identifiant d’un autre Lien (cf. figure 1b), ou indique parfois l’identifiant du Lien lui-même.

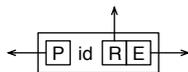


Fig. 1a : le Lien

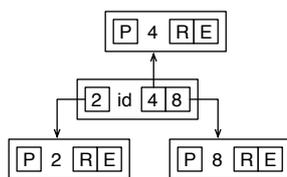


Fig. 1b : le Lien référence des Liens

Aucun Lien ou groupe de Liens n’est séparé. Tous les Liens sont directement ou indirectement *reliés*, ils constituent un ensemble fortement connexe.

Il ne faut pas confondre le Lien avec un triplet RDF du Web sémantique [9]. D’abord, le Lien forme un triplet uniquement identifié, c’est à dire un *quadruplet*. Ensuite, le Web sémantique est un Web de données qui représente la connaissance par *une pluralité de triplets* référençant *une pluralité de ressources* (les *choses* du monde) via des URI (Uniform Resource Identifier). Le procédé, lui, représente la connaissance par *une pluralité de Liens* référençant des Liens uniquement identifiés.

RDF traite des données (les ressources et littéraires) *et* des références sur des données (les triplets), alors que le procédé ne traite que des Liens.

3.2 Base de Liens

La totalité des contenants et des contenus informatifs est représentée de façon homogène à l'aide de Liens. Le procédé manipule donc une seule structure de donnée, de taille fixe, ce qui permet d'**accélérer l'archivage et l'accès à l'information** :

On accède à l'information par un *ensemble restreint de requêtes élémentaires*, optimisées pour interroger une *Base de Liens* (cf. Illustration 3, p. 43). Tous les Liens d'une Base de Liens sont uniques, *identifiés et indexés*. Le Lien, à lui seul, peut déduire les Liens dont il a besoin pour exprimer l'information qu'il représente. Ainsi, il est possible de récupérer le lot entier des Liens concernant une information, en *une seule transaction*. En outre, il est aisé de détecter les Liens statiques qui ne sont jamais, ou que rarement modifiés. Ces Liens statiques peuvent être répliqués sur les machines clientes afin de réduire les transactions aux *seules informations dynamiques*. Par ailleurs, le procédé peut donner à l'information une *validité temporelle* (cf. 3.3). On évite ainsi les recharges inutiles de Liens déjà connus, ou d'informations distantes encore valides.

Un Lien est uniquement identifié dans le contexte d'une ou plusieurs Bases de Liens. Ces Bases de Liens peuvent résider sur une machine isolée, ou sur plusieurs machines reliées en réseau, par une topologie quelconque. Les questions relatives à l'optimisation ou à la synchronisation des Bases de Liens, à la génération et à la distribution d'identificateurs uniques (et universels [10]), ou à la communication d'informations représentées par des Liens, ne sont pas traitées dans cet article. Il est important de comprendre que :

le Lien représente l'*unité élémentaire d'archivage* du procédé.

3.3 Champs supplémentaires du Lien

La structure du Lien peut contenir des champs supplémentaires. Ainsi, des *attributs temporels* (cf. sous-section 8.1) donnent une période de validité au Lien (date d'activation, date de désactivation, ...). Doter les Liens d'attributs temporels permet la mise en oeuvre d'une *gestion temporelle native* de l'information, inhérente et efficace, placée au coeur même de la totalité de l'information traitée par le procédé. En d'autres termes, c'est rendre persistant, aussi *simplement* que possible, l'ensemble des modifications apportées à l'intégralité d'un système d'informations et des agencements qu'il maintient, aussi *complexes* soit-ils.

Ces champs supplémentaires peuvent également servir à contenir des informations de typage, de catégorisation ou encore des décorations liées aux droits d'accès, au mode de modification (lecture seule), etc. D'autres champs peuvent être réservés à l'implémentation de techniques relatives au *classement* de l'information (fréquence d'utilisation, indice de classement), à la *logique floue* (degré de confiance de l'information, coefficient indiquant l'exactitude du Lien), aux *réseaux neuronaux* (poids synaptique des Liens)...

Dans le cadre de cet article, seule la structure minimale du Lien, composée de quatre champs (id, P, R, E), est présentée.

4 Vocabulaire

“Un enfant ayant atteint un niveau d'élocution critique dans sa langue maternelle pourra enrichir par lui-même et à pas de géant son vocabulaire. L'enfant peut se servir du langage pour étendre son langage.” [6] La difficulté était d'*amorcer* le langage, d'apprendre les premiers mots, les premiers concepts.

Les termes qui constituent le vocabulaire du procédé débutent tous par une lettre majuscule, ceci afin de ne pas les confondre avec la définition qui leur est habituellement attribuée, et pour les identifier plus facilement comme constituants du procédé.

4.1 Quatre termes principaux

Le vocabulaire s'articule autour de quatre termes principaux : *Contenant*, *Contenu*, *Relation* et *Lien*. Une compréhension globale de ces quatre premiers termes est suffisante dans un premier temps. Le Lien étant l'unité d'archivage du procédé, on peut déjà introduire le fait que les Contenants, les Contenus et les Relations, tous sont identifiés et représentés par des Liens.

D'autres termes, comme *ContenantDesRelations*, *ContenuParent*, *RelationParent*, *LienDeContenance*, etc, sont dérivés des quatre termes principaux. Leur sens est souvent implicite, et on trouvera leur description précise en se référant ultérieurement à cette section (cf. 4.3).

4.2 Notation Contenant-Relation

Les représentations schématiques que l'on retrouve dans la majorité des figures, utilisent une notation *Contenant-Relation*, à ne pas confondre avec une notation entité-relation³. Elle est développée et expliquée parallèlement à la définition des termes (cf. “noté selon la figure x”, sous-section 4.3). Cependant, elle est reprise sommairement au début de la section suivante (5. Un Modèle simple).

4.3 Définition des termes

Les définitions des termes présentés ci-dessous sont récursives : un terme A s'explique avec un terme B, lequel s'explique directement ou indirectement grâce au terme A. Au fur et à mesure que le concept décrit par ce “langage” prendra corps, la définition de ces termes deviendra de plus en plus claire. Il est conseillé de poursuivre la lecture des sections suivantes et de consulter cette section-ci autant de fois que nécessaire, plutôt que de vouloir comprendre le procédé en étudiant dans le détail les termes utilisés et toutes les interdépendances possibles.

Les références à des termes définis plus en avant sont soulignées. Dans une première lecture, il suffit de sauter les parties grisées du texte.

- **Contenu** : Un Contenu (noté selon la figure 2, p. 11) représente *un élément distinct* parmi une collection d'éléments, laquelle collection est un Contenant. Un Contenu est identifié et représenté par un Lien (noté selon la figure 3, p. 11). Plus précisément, un Contenu est identifié et représenté par le LienDeContenance d'une RelationDeContenance 'RC'.



Fig. 2 : un Contenu



Fig. 3 : un Contenu est un Lien

- **Contenant** : Un Contenant (noté selon la figure 4) représente *une collection d'éléments*, lesquels éléments sont des Contents. Un Contenant contient des Contents (figure 5). Un Contenant est toujours le Contenu d'un Contenant (figure 6), donc un Contenant est aussi un Contenu. Un Contenant peut être contenu par lui-même (figure 7, p. 12). Puisqu'un "Contenant est aussi un Contenu", il est donc également identifié et représenté par un Lien (noté selon la figure 8, p. 12).

Plus précisément, un Contenant est identifié et représenté par un LienDeContenance de la RelationDeContenance 'RC' de son propre Contenant.

Un Contenant est *défini* par l'ensemble des Relations (les RelationParents) qui associent ce Contenant (ContenantParent) à d'autres Contents ou à lui-même (ContenantEnfant).

Chaque Contenant nécessite au moins une RelationParent particulière, appelée RelationDeContenance 'RC' dont les LienDeContenance serviront à identifier et représenter les Contents de ce Contenant.

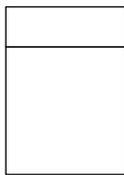


Fig. 4 : un Contenant

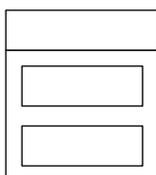


Fig. 5 : un Contenant et deux Contents

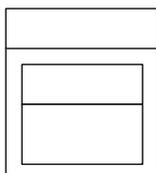


Fig. 6 : un Contenant est un Contenu

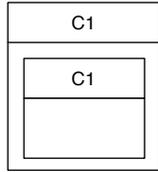


Fig. 7 : un Contenant contenu par lui-même



Fig. 8 : un Contenant est un Lien

- **Lien** : Un Lien est un Contenu d'une Relation. Chaque terme décrit dans cette section est identifié et représenté par un Lien. Le Lien constitue une structure de donnée dotée d'au moins quatre champs :
 - id, identifiant unique du Lien (noté selon la figure 9).
 - R, référence sur un Contenant (notée selon la figure 10), ce Contenant est la Relation qui contient le Lien.
 - P et E, références dont la vocation est de **relier deux Contents** (notées selon la figure 11). La référence P indique le ContenuParent 2, la référence E indique le ContenuEnfant 3.

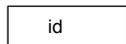


Fig. 9 : un Lien est identifié

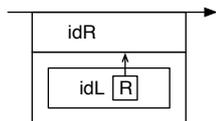


Fig. 10 : la référence R du Lien, $R(idL) = idR$

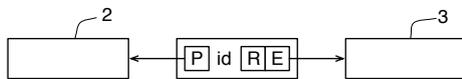


Fig. 11 : le Lien relie deux Contents via ses références P et E

- **Relation** : Une Relation (notée selon la figure 12) est un Contenant.

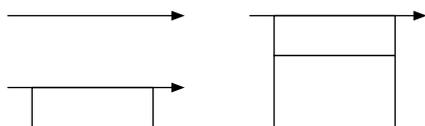


Fig. 12 : les différentes notations d'une Relation

Comme un Contenant, une Relation est également un Contenu (figure 13). Elle est donc également identifiée et représentée par un Lien (figure 14), plus précisément par un LienDeContenance d'une RelationDeContenance 'RC'. Comme un Contenant, une Relation contient des Contents. Mais une Relation est un *Contenant particulier* (noté selon figure 15), parce que ses Contents sont des Liens, les Liens dont la référence R indique ladite Relation (ce qui n'est pas le cas des Contents d'un Contenant).

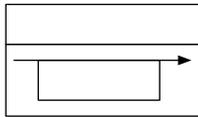


Fig. 13 : une Relation est un Contenu

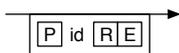


Fig. 14 : une Relation est un Lien

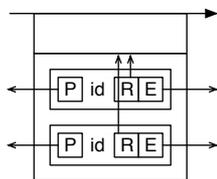


Fig. 15 : une Relation est un Contenant

Une Relation a la faculté d'être contenue par elle-même (figure 16), de sorte que la référence R indiquant sa Relation correspond à son propre identifiant id (cf. sous-section 6.9 Réflexivité).

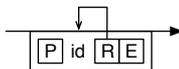


Fig. 16 : une Relation qui se contient elle-même

Les Relations sont contenues dans un Contenant (figure 17), appelé ContenantDesRelations 'CR'.

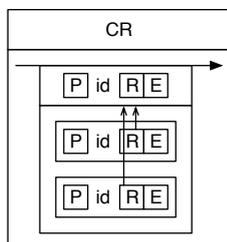


Fig. 17 : le ContenantDesRelations 'CR'

La vocation d'une Relation est d'*associer deux Contents* (notés selon figure 18, p. 14). Une Relation a un *sens*, elle associe un ContenantParent 4 à un ContenantEnfant 5.

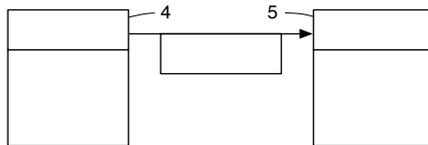


Fig. 18 : une Relation associe un ContenantParent à un ContenantEnfant

Une Relation étant également un Contenant, il est possible que l'un ou l'autre, ou les deux Contenants associés par une Relation puissent être aussi une Relation. L'exemple illustré par la figure 19 montre une Relation 'R' dont le ContenantParent est une Relation, la Relation 'Rp', et dont le ContenantEnfant associé est un Contenant, le Contenant 'Ce'.

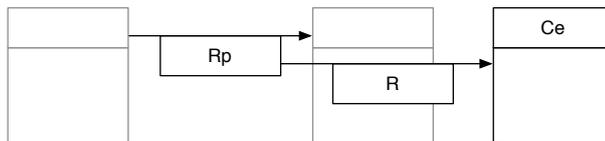


Fig. 19 : une Relation 'R' associant une Relation 'Rp' à un Contenant 'Ce'

IMPORTANT : les Liens (figure 20) 6 d'une Relation 7 *relie deux Contents* : un ContenuParent 8 du ContenantParent 9 à un ContenuEnfant 10 du ContenantEnfant 11. C'est ainsi que tous les Liens sont câblés.

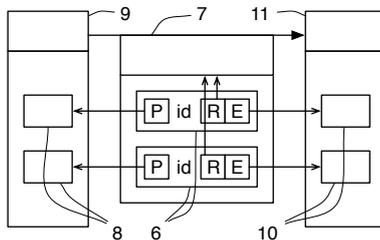


Fig. 20 : le Lien et ses références P, R et E

- **ContenantParent** : Le ContenantParent est le Contenant *source* (figure 18) 4 d'une Relation.
- **ContenantEnfant** : Le ContenantEnfant est le Contenant de *destination* (figure 18) 5 d'une Relation.
- **ContenuParent** : Le ContenuParent est le Contenu (figure 20) 8 indiqué par la *référence P* du Lien 6.
- **ContenuEnfant** : Le ContenuEnfant est le Contenu (figure 20) 10 indiqué par la *référence E* du Lien 6.
- **RelationParent** : On appelle les RelationParents d'un Contenant (figure 21, p. 15) 12, les Relations 13 dont le ContenantParent désigne ce Contenant.

Un Contenant 12 est *défini* par une ou plusieurs RelationParents 13.

Une Relation désigne son ContenantParent par la Relation 'R-CP' du ContenantDesRelations 'CR'.

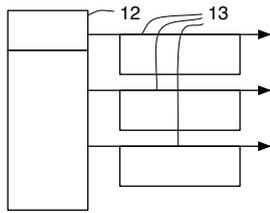


Fig. 21 : les RelationParents

- **RelationEnfant** : On appelle les RelationEnfants d'un Contenant (figure 22) 14, les Relations 15 dont le ContenantEnfant désigne ce Contenant.

Une Relation désigne son ContenantEnfant par la Relation 'R-CE' du ContenantDesRelations 'CR'.

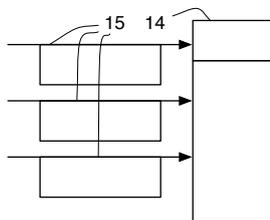


Fig. 22 : les RelationEnfants

- **ContenantDesRelations 'CR'** : Toutes les Relations sont contenues par le ContenantDesRelations 'CR' (figure 23) 16. Il sert à définir les propriétés des Relations, notamment le ContenantParent et le ContenantEnfant de chaque Relation.

Comme tout Contenant, le ContenantDesRelations est défini par ses RelationParents, il est défini de façon *récursive* par des RelationParents 18 qu'il sert justement à définir 17. Le ContenantDesRelations est défini par au moins deux RelationParents : la Relation 'R-CP' et la Relation 'R-CE'.

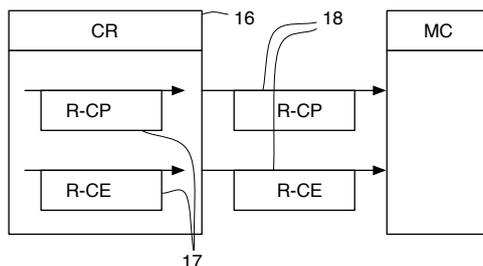


Fig. 23 : définition récursive des Relations

- **MétaContenant 'MC'** : Le MétaContenant 'MC' (cf. figures 23, 24, 27; sous-section 6.2 p. 26) est le *Contenant des Contenants*, il se contient lui-même. Il identifie un Système.
- **Relation 'R-CP'** : Comme toutes les Relations, la Relation 'R-CP' (figure 23) 17 est un Contenu du ContenantDesRelations ('CR' 16), mais elle est aussi la RelationParent ('R-CP' 18) du ContenantDesRelations 'CR' servant à *désigner le ContenantParent* de chaque Relation, y compris d'elle-même.

Le ContenantParent désigné, est contenu par le MétaContenant 'MC' qui contient tous les Contenants. Ainsi les Liens de la Relation 'R-CP' relient deux Contenus : une Relation du ContenantDesRelations 'CR' à un Contenant (le ContenantParent) du MétaContenant 'MC'.

Par exemple, la Relation 'R-CP' (figure 27, p. 21) relie par son Lien 'e' la Relation 'prénom' (réf. P) au Contenant 'Personnes' (réf. E), lequel désigne le ContenantParent de la Relation 'prénom'.

- **Relation 'R-CE'** : Comme toutes les Relations, la Relation 'R-CE' (figure 23, p. 15) 17 est un Contenu du ContenantDesRelations ('CR' 16), mais elle est aussi la RelationParent ('R-CE' 18) du ContenantDesRelations 'CR' servant à *désigner le ContenantEnfant* de chaque Relation, y compris d'elle-même.

Le ContenantEnfant désigné, est contenu par le MétaContenant 'MC' qui contient tous les Contenants. Ainsi les Liens de la Relation 'R-CE' relient deux Contenus : une Relation du ContenantDesRelations 'CR' à un Contenant (le ContenantEnfant) du MétaContenant 'MC'.

Par exemple, la Relation 'R-CE' (figure 27, p. 21) relie par son Lien 'g' la Relation 'prénom' (réf. P) au Contenant 'Prénoms' (réf. E), lequel désigne le ContenantEnfant de la Relation 'prénom'.

- **LienDeContenance** : Les LienDeContenance sont les Liens contenus par une RelationDeContenance 'RC', c'est à dire les Liens (figure 24) 22 dont la référence R indique une RelationDeContenance 'RC' 20.

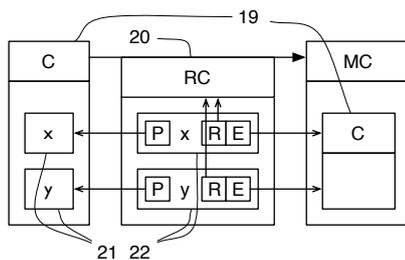


Fig. 24 : les LienDeContenance (22) de la RelationDeContenance 'RC'

- **RelationDeContenance 'RC'** : Chaque Contenant 'C' (figure 24) est au moins défini par une RelationParent particulière, appelée RelationDeContenance 'RC', ainsi nommée parce qu'elle désigne ce qui peut être contenu dans les limites d'un Contenant. La RelationDeContenance 'RC' est particulière parce que ses LienDeContenance *identifient et représentent les Contenus* de ce Contenant 'C'. Les Contenus 'x' et 'y' 21 représentés graphiquement en tant que Contenus d'un Contenant 'C' *sont* les LienDeContenance 'x' et 'y' 22 représentés graphiquement en tant que Liens d'une Relation 'RC'.

Le ContenantEnfant ('MC') de la RelationDeContenance ('RC') d'un Contenant ('C') est le Contenant de ce dernier ('C' est contenu par 'MC').

Les LienDeContenance (figure 24) 22 de la RelationDeContenance 'RC' 20 d'un Contenant 'C' 19 servent à *identifier et représenter les Contenus* 21 de ce Contenant 'C', et à les *relier à leur Contenant* 19. Ainsi les Contenus 21 *sont* les LienDeContenance 22 dans une représentation graphique différente. Le LienDeContenance 'x' identifie et représente le Contenu 'x' ($P(x) = id(x)$). Par sa référence E, ce LienDeContenance 'x' relie le Contenu 'x' qu'il représente à son Contenant 'C' ($E(x) = id(C)$).

4.4 Autres définitions

- **Modèle** : Un Modèle est un ensemble de Contenants associés par des Relations, et ayant pour objectif de structurer l'information.

Les Relations étant également des Contenants, on peut donc créer des Modèles (figure 19, p. 14) où une Relation 'R' associe, par exemple, une Relation 'Rp' à un Contenant 'Ce'. Mais le propos de cet article n'est pas d'explorer les possibilités de modélisation du procédé. On retiendra que le procédé organise l'information en créant des Modèles.

- **Système** : Un Système est un système de traitement d'informations mis en œuvre par le procédé et basé sur le traitement de Liens. Un Système possède au moins les propriétés suivantes :
 - il est *auto-décrit*, il n'est pas décrit par des éléments qui sont extérieurs à lui-même,
 - il est *réflexif*, il se contient et se manipule lui-même, il s'auto-représente et décrit la connaissance qu'il a de lui-même par un Modèle,
 - il est *autoporteur*, comme il peut enrichir et améliorer la définition des Modèles qu'il permet de créer, il peut également enrichir et améliorer la définition de son propre Modèle,
 - il est *identifié* par un Lien, de sorte que ce Lien pourrait être un Lien utilisé par un Méta-Système...
- **Noyau** : Un Noyau est un Modèle particulier qui permet la définition et l'amorçage d'un Système. Autrement dit, un Noyau représente l'agencement minimal d'un ensemble de Liens nécessaire à la création d'un Système auto-décrit, réflexif, autoporteur et identifié à l'aide de Liens.
- **Hyper Information** : Toute l'information traitée par le procédé est identifiée et représentée par des Liens. Ces Liens sont câblés de sorte que chacun d'eux peut retrouver le contexte dans lequel l'information qu'ils représentent est utilisée, il n'y a plus de segmentation sémantique (cf. 1.4, p. 5). L'information peut retrouver la *méta-information* utile à sa description, et/ou peut connaître l'*intra-information* qu'elle sert à décrire.

Le procédé traite de l'*Hyper Information*, ce terme pour préciser que toute l'information traitée par le procédé est *reliée* et *sémantiquement marquée*.

5 Un Modèle simple

A l'aide d'un exemple simple (figure 25), cette section décrit d'une manière générale comment l'information est modélisée par le procédé. Elle permet en outre de se familiariser avec la vocabulaire utilisé et présenté dans la section précédente.

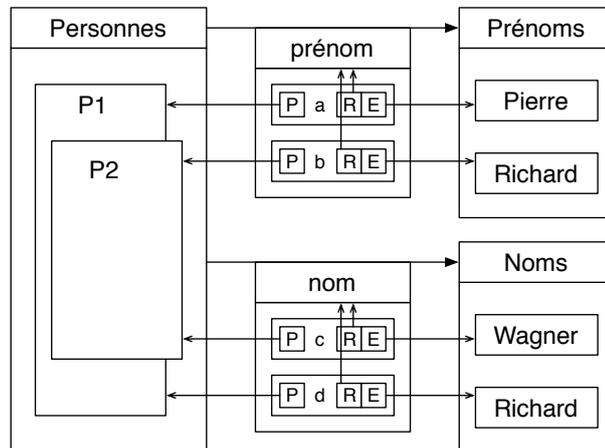


Fig. 25 : un Modèle simple

5.1 Le Modèle 'Personnes'

La notation graphique Contenant-Relation utilisée pour expliquer ce procédé est illustrée par un simple exemple (figure 25) qui *définit* un Contenant 'Personnes' par une Relation avec un Contenant 'Prénoms' et une Relation avec un Contenant 'Noms'.

Le Modèle de la figure 25 est constitué de plusieurs éléments :

- trois Contenants :
 - 'Personnes',
 - 'Prénoms',
 - 'Noms',
- six Contents :
 - 'P1' et 'P2', contenus du Contenant 'Personnes',
 - 'Pierre' et 'Richard', contenus du Contenant 'Prénoms',
 - 'Wagner' et 'Richard', contenus du Contenant 'Noms',
- deux Relations :
 - 'prénom',
 - 'nom',
- quatre Liens :
 - Liens 'a' et 'b', contenus de la Relation 'prénom',
 - Liens 'c' et 'd', contenus de la Relation 'nom'.

Dans un premier temps, on ne tiendra pas compte des RelationDeContenance 'RC', elles ne sont pas dessinées dans cet exemple. Une RelationDeContenance 'RC' est obligatoire pour chaque Contenant, ses LienDeContenance servant à identifier et représenter les Contenus d'un Contenant.

5.2 Un Contenant est défini par ses RelationParents

Le Contenant '*Personnes*' (figure 25, p. 18) est *défini* par deux RelationParents que sont 'prénom' et 'nom'. Les Contenants 'Prénoms' et 'Noms' n'ont pas de RelationParents, ils représentent simplement des énumérations de prénoms et de noms. Par contre, les Contenants 'Prénoms' et 'Noms' comportent chacun une RelationEnfant : le Contenant 'Prénoms' a comme RelationEnfant la Relation 'prénom', et le Contenant 'Noms' a comme RelationEnfant la Relation 'nom'. Ainsi le Contenant 'Noms' par exemple, peut indirectement déduire qu'il sert à définir le Contenant 'Personnes'. Comment un Contenant peut-il déduire ses RelationParents et ses RelationEnfants ? La réponse est détaillée un peu plus en avant, dans les sous-sections 5.6 et 5.7.

5.3 Création d'un nouveau Contenu

Créer un nouveau Contenu, équivaut à créer un Lien pour chacune des RelationParents du Contenant. Ainsi, créer une nouvelle personne 'P1' (figure 25, p. 18) revient à créer un Lien pour les Relations 'prénom' et 'nom', soit les Liens 'a' et 'd'.

Il s'agit là d'une règle générale. La création d'un nouveau Contenu peut dépendre de propriétés relatives aux Relations. Par exemple : • est-ce qu'il est possible de relier la personne 'P1' à plusieurs prénoms (cf. sous-section 5.4) ? • comment sont gérés les attributs de cardinalité des Relations ? • peut-on omettre de définir un Lien pour une RelationParent ou est-ce obligatoire ? • y a-t-il un ordre de renseignement des RelationParents ? • comment sont traitées les contraintes d'intégrité référentielle ⁴ ? La sous-section 7.6 montre comment *enrichir* les Relations de telles propriétés, mais ces questions ne font pas directement partie du propos de cet article, qui se veut orienté sur la construction d'un Noyau. Lorsque le lecteur aura lu ce document dans son intégralité, il pourra apporter lui-même ses propres réponses.

On dit qu'un Contenu *hérite ses propriétés* des RelationParents de son Contenant. On dit qu'un Contenant est *défini* par ses RelationParents. Ainsi le Contenu 'P2' hérite la propriété 'Richard' par la Relation 'prénom' et la propriété 'Wagner' par la Relation 'nom'. Le Contenant 'Personnes' est défini par deux RelationParents : la Relation 'prénom' et la Relation 'nom'.

5.4 Types de Relation

On suppose pour l'instant, que par l'intermédiaire des Liens d'une Relation, un ou *plusieurs* ContenuParents peuvent être reliés à *un* seul ContenuEnfant. Autrement dit, on suppose que les Relations utilisées dans ce document sont, par défaut, des Relations du type "*de plusieurs vers un*". Par exemple, *plusieurs* Personnes peuvent avoir *un* même Prénom. On notera quatre types de Relation possibles, une Relation :

- **de un vers un** (figure 26A), dont le ContenuParent et le ContenuEnfant référencés par chacun de ses Liens sont obligatoirement différents,

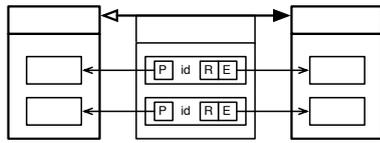


Fig. 26A : Relation "de un vers un"

- **de un vers plusieurs** (figure 26B), dont le ContenuParent de chacun de ses Liens est quelconque et dont le ContenuEnfant est obligatoirement différent,

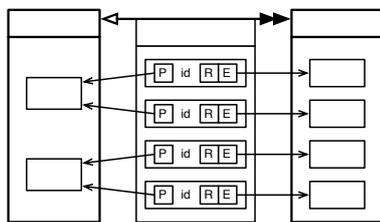


Fig. 26B : Relation "de un vers plusieurs"

- **de plusieurs vers un** (figure 26C), dont le ContenuParent de chacun de ses Liens est obligatoirement différent et dont le ContenuEnfant est quelconque,

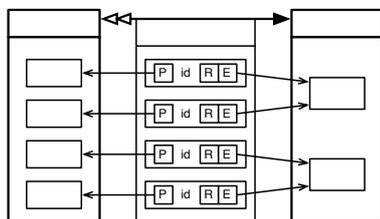


Fig. 26C : Relation "de plusieurs vers un"

- **de plusieurs vers plusieurs** (figure 26D), dont le ContenuParent et le ContenuEnfant de chacun de ses Liens sont quelconques.

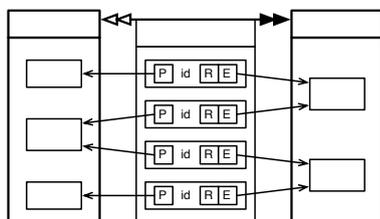


Fig. 26D : Relation "de plusieurs vers plusieurs"

5.5 Modélisation des Relations

Les Relations 'prénom' et 'nom' de la figure 25 (p. 18) sont des Contenants particuliers. Elles sont des Contenants particuliers parce que leurs Contents sont des Liens ('a', 'b' et 'c', 'd'), plus précisément, les Liens dont la référence R indique l'identifiant id des Relations 'prénom' et 'nom'. Mais les Relations sont également des Contents, et elles sont contenues dans un Contenant appelé le ContenantDesRelations, noté 'CR' (figure 27).

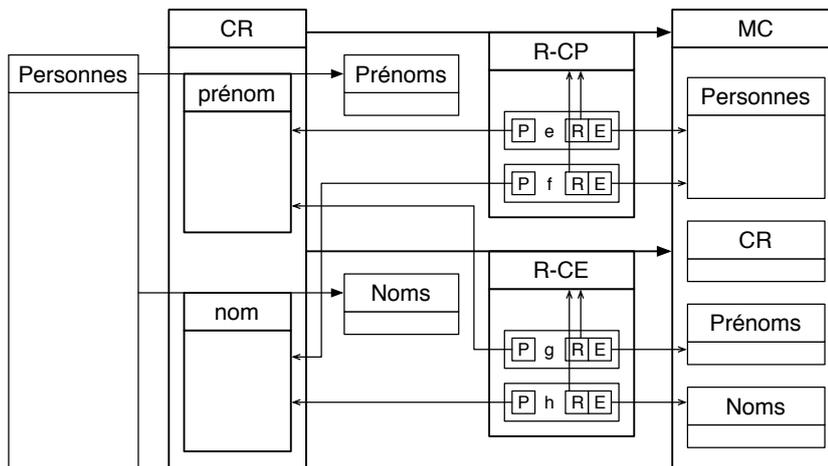


Fig. 27 : modélisation des Relations 'prénom' et 'nom'

Au même titre qu'un Contenu hérite ses propriétés de son Contenant qui lui-même est défini par ses RelationParents, de même une Relation hérite ses propriétés du ContenantDesRelations 'CR' qui lui-même est défini par ses RelationParents, que sont les Relations 'R-CP' et 'R-CE'.

La vocation d'une Relation est d'associer un ContenantParent à un ContenantEnfant, il faut donc au moins désigner pour chaque Relation quel est son ContenantParent et quel est son ContenantEnfant. Ceci est réalisé par les deux RelationParents du ContenantDesRelations 'CR' que sont les Relations 'R-CP' (ContenantParent) et 'R-CE' (ContenantEnfant).

Pour la Relation 'prénom' :

- Le **Lien 'e'** désigne le Contenant 'Personnes' comme *ContenantParent de la Relation 'prénom'*, autrement dit, le Contenu 'prénom' hérite la propriété 'Personnes' par le Lien 'e' de la Relation 'R-CP',
- le **Lien 'g'** désigne le Contenant 'Prénoms' comme *ContenantEnfant de la Relation 'prénom'*, autrement dit, le Contenu 'prénom' hérite la propriété 'Prénoms' par le Lien 'g' de la Relation 'R-CE'.

Pour la Relation 'nom' :

- Le **Lien 'f'** désigne le Contenant 'Personnes' comme le *ContenantParent de la Relation 'nom'*,
- le **Lien 'h'** désigne le Contenant 'Noms' comme *ContenantEnfant de la Relation 'nom'*.

5.6 Dédution des RelationParents

Chaque Contenant peut déduire, par l'intermédiaire de ses RelationParents, les Contenants qu'il *utilise pour se définir*.

Un Contenant peut déduire ses RelationParents en interrogeant la Relation 'R-CP' du ContenantDesRelations 'CR'. Par exemple, le Contenant 'Personnes' (figure 27, p. 21) déduit ses RelationParents en interrogeant l'ensemble des Liens de la Relation 'R-CP' dont la référence E indique l'identifiant id qui représente le Contenant 'Personnes'. Le résultat de la requête retourne les Liens 'e' et 'f'. La référence P de chacun de ces Liens indique une RelationParent : la référence P(e) indique la Relation 'prénom' et la référence P(f) indique la Relation 'nom'.

Connaissant ses RelationParents, un Contenant peut déduire pour chacune d'elles le ContenantEnfant associé. Par exemple, la RelationParent 'prénom' déduit son ContenantEnfant en interrogeant l'ensemble des Liens de la Relation 'R-CE' dont la référence P indique l'identifiant id qui représente cette RelationParent 'prénom'. Le résultat de la requête retourne le Lien 'g'. La référence E de ce Lien indique le ContenantEnfant recherché, c'est à dire le Contenant 'Prénoms'.

5.7 Dédution des RelationEnfants

Chaque Contenant peut déduire, par l'intermédiaire de ses RelationEnfants, les Contenants qu'il *sert à définir*.

Un Contenant peut déduire ses RelationEnfants en interrogeant la Relation 'R-CE' du ContenantDesRelations 'CR'. Par exemple, le Contenant 'Noms' (figure 27, p. 21) déduit ses RelationEnfants en interrogeant l'ensemble des Liens de la Relation 'R-CE' dont la référence E indique l'identifiant id qui représente le Contenant 'Noms'. Le résultat de la requête retourne les Liens dont la référence P indique une RelationEnfant, en l'occurrence la requête retourne le Lien 'h', et la référence P(h) indique la RelationEnfant 'nom'.

Connaissant ses RelationEnfants, un Contenant peut déduire pour chacune d'elles le ContenantParent associé. Par exemple, la RelationEnfant 'nom' déduit son ContenantParent en interrogeant l'ensemble des Liens de la Relation 'R-CP' dont la référence P indique l'identifiant id qui représente cette RelationEnfant 'nom'. Le résultat de la requête retourne le Lien 'f'. La référence E de ce Lien indique le ContenantParent recherché, c'est à dire le Contenant 'Personnes'.

5.8 Modélisation des Contenants

Les Contenants 'Personnes', 'Prénoms', 'Noms' et 'CR' (figure 27, p. 21) sont également des Contenus, ils sont contenus par le MétaContenant 'MC'. Et ce MétaContenant 'MC' est lui-même défini par ses propres RelationParents (non représentées dans la figure 27).

Les méta-informations (les Contenants) peuvent devenir à leur tour des informations (des Contenus), lesquelles sont définies par d'autres méta-informations (d'autres Contenants), etc. Les Contenants 'Prénoms' et 'Noms' peuvent devenir les Contenus d'un autre Contenant ('Noms propres', par exemple). Le Procédé autorise la *généralisation* de l'information sur un nombre indéfini de niveaux, "des niveaux de description supérieurs ajoutant des connaissances sur des niveaux inférieurs". [11]

5.9 Tout est Lien

Il est opportun de rappeler ici ce qui caractérise le procédé :

Il s'agit d'un procédé permettant la création de Systèmes de traitement d'informations, dans lesquels le Système lui-même, ainsi que les contenants et les contenus informatifs traités, sont créés, identifiés, archivés, modélisés, communiqués, modifiés et supprimés; par la création et la modification de Liens, ou parfois par leur suppression.

La totalité de l'information traitée et de l'information servant à décrire le Système est archivée sous la forme d'une seule structure de donnée : le Lien. La section suivante (6. Construction d'un Noyau) montre comment il est possible d'initier un tel Système.

Ainsi, l'ensemble des éléments du Modèle simple illustré par la figure 25 (p. 18), est totalement et uniquement identifié et représenté par des Liens : les trois Contenants, les six Contenus, les deux Relations, et bien sûr les quatre Liens, sont tous des Liens (figure 28). On notera que *seuls* les Contenus des deux Relations, c'est à dire les Liens 'a', 'b' et 'c', 'd', ont leur référence R qui indique directement leur Contenant, c'est à dire une Relation.

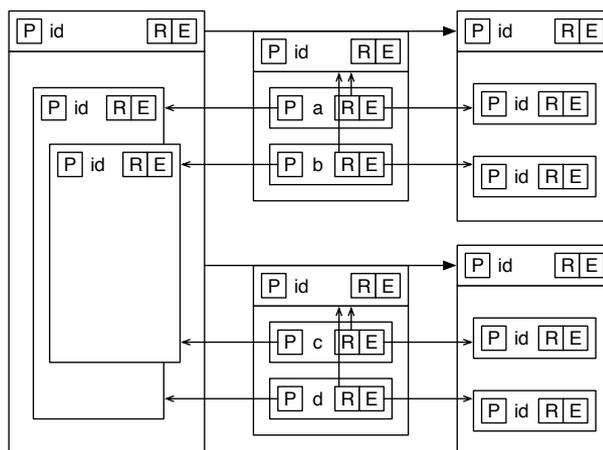


Fig. 28 : tous les éléments du Modèle de la fig. 25 sont identifiés et représentés par des Liens

5.10 Le « modèle relié »

Le modèle de données proposé par ce procédé est nommé *modèle relié*. Aujourd'hui les arrangements de l'information sont régis par la *segmentation* et la *duplication* des informations, au contraire le modèle relié propose des arrangements où l'information est *reliée* et *identifiée*.

Avec la représentation tabulaire du modèle relationnel, l'enregistrement 'Richard' de la colonne 'Prénom' de la table 'Personne' ne constitue qu'une représentation typographique passive et déconnectée. L'information 'Richard' est *segmentée* de sa propre méta-information, elle ne peut pas déduire qu'elle est un prénom utile à décrire une personne.

Avec le *modèle relié*, on dit que *l'information a connaissance d'elle-même*. Le Lien qui identifie l'information 'Richard' n'est pas segmenté des Liens qui identifient les méta-informations 'Prénom' et 'Nom'. Si on saisit 'Café Richard' et 'Richard International' dans un Système, l'information 'Richard' pourra dé-

duire (sans autre opération supplémentaire) qu'elle sert également à définir le 'nom' d'un 'Restaurant' ou d'une 'Société de transport'.

Le Lien, à lui seul, peut retrouver la totalité des contextes dans lesquels il est utilisé, et du Système dont il fait partie. Les Liens sont câblés de manière à ce que chacun d'eux permet la reconstruction du Système dont il a besoin pour donner corps à l'information qu'il représente. Le modèle relié *ne segmente pas* les informations des méta-informations, mais les *relie*. Il donne à *toute information* la possibilité de déduire (de connaître) l'ensemble des Liens lui permettant de se définir et, en accord avec l'axiome proposé (cf. 2.2), lui permettant d'*exister*.

L'information est intrinsèquement reliée à son sens sémantique, et ceci sur autant de niveaux de méta-information désirés. Toute information ('Richard') a connaissance des informations *qui l'utilisent* (les Personnes 'Richard Wagner' et 'Pierre Richard'), de quels Contenants *elle fait partie* ('Prénoms' et 'Nom'), et des informations avec lesquelles *elle est constituée* (les Lettres 'R, i, c, h, a, r, d'). ***Le modèle relié donne une nouvelle dimension à l'agencement et à la navigation de l'information***, il traite de l'Hyper Information (cf. 4.4).

6 Construction d'un Noyau

“Nos pensées sur nos expériences de pensées sont elles-mêmes des expériences de pensées.” [11]

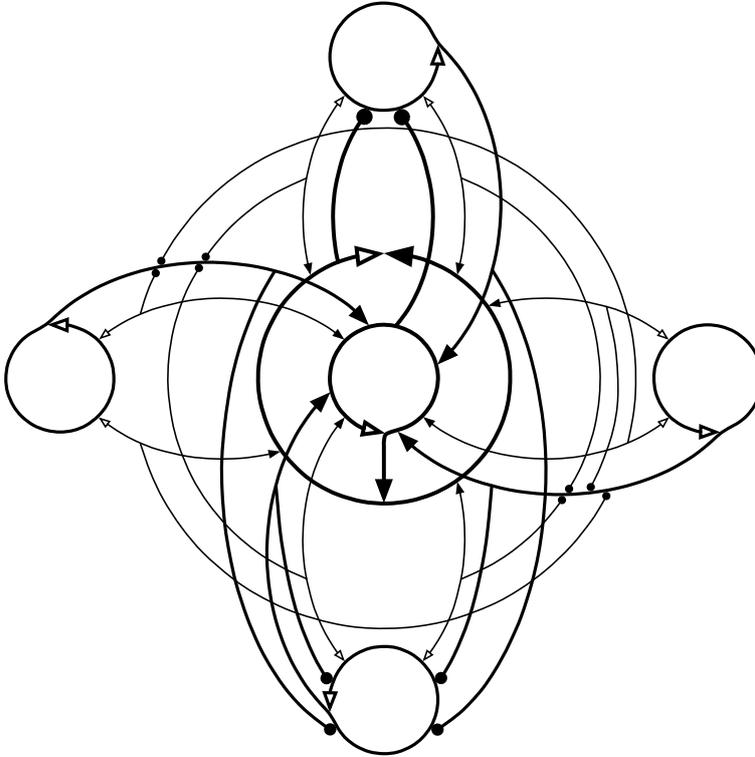


Illustration 2 : câblage du Noyau

6.1 L'amorçage

Un Noyau représente le minimum d'information nécessaire à l'amorçage d'un Système. Le Noyau proposé dans cet article a simultanément besoin d'un ensemble de *quatorze Liens* pour être cohérent, cependant sa construction est expliquée ci-après par un récit chronologique, où les quatorze Liens sont ajoutés l'un après l'autre. Avant de commencer, il peut être utile de rappeler l'axiome sur lequel se fonde le procédé (cf. 2.2) :

Un objet n'a pas d'existence dans l'absolu,
tout objet *existe par ses Liens* avec d'autres objets.
Ainsi, de façon récursive, il n'y a plus que des Liens.

Ci-dessous les étapes de création d'un Noyau de quatorze Liens. Les figures 29 à 36 (p. 26 à 32) montrent progressivement la construction de ce Noyau, chaque figure est commentée dans les sous-sections suivantes. Bien qu'un modèle formel s'explique habituellement à partir de son niveau *atomique*, ici le Lien, l'explication suivante part du *global*, du Contenant des Contenants.

6.2 Le Contenant des Contenants

Aucun élément permettant la construction du Noyau n'est extérieur au Noyau. Un Noyau est *auto-décrit*. Tous les éléments sont donc inclus dans un contenant. Ce contenant est le Contenant des Contenants, il est appelé **MétaContenant 'MC'** (figure 29). Le MétaContenant 'MC' n'est pas extérieur à lui-même, il contient tout, c'est à dire lui-même, il est donc à la fois le Contenant 23 et le Contenu 24.

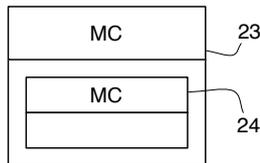


Fig. 29 : le MétaContenant 'MC'

6.3 La première Relation

Mais le MétaContenant 'MC' n'a pas encore "connaissance de lui-même" : il ne peut pas déduire qu'il est son propre Contenant et qu'il se contient. Il n'*existe* pas encore, car selon l'axiome énoncé, toute information n'existe qu'en *lien* avec d'autres informations. Un ensemble de Liens constitue une Relation. La seule RelationParent possible, permettant de définir le MétaContenant 'MC', est la Relation que le MétaContenant 'MC' peut avoir avec lui-même. La figure 30A illustre la première Relation, elle est appelée **Relation 'RC-MC'**, c'est une RelationDeContenance.

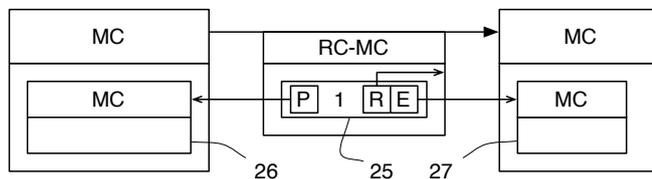


Fig. 30A : le premier Lien '1' du Noyau de quatorze Liens

6.4 Le premier Lien

Une RelationDeContenance associe deux Contenants, le ContenantEnfant qu'elle associe au ContenantParent est le Contenant de ce dernier. Autrement dit et à l'aide de la figure 24 (p. 16), le ContenantEnfant 'MC' que la RelationDeContenance 'RC' associe au ContenantParent 'C' est le Contenant du ContenantParent 'C' (le Contenant du ContenantParent 'C' étant le Contenant 'MC'). Un LienDeContenance relie un Contenu (référence P) à son Contenant (référence E). Ainsi, **le premier Lien 25** (figure 30A) du Noyau est un LienDeContenance, il relie le Contenu 'MC' 26 (référence P) à son Contenant 'MC' 27 (référence E), c'est à dire au MétaContenant 'MC'.

Ainsi, le MétaContenant 'MC' prend "connaissance de lui-même" : il n'est plus déconnecté de lui-même et peut maintenant déduire que son Contenant est lui-même et qu'il se contient. Selon l'axiome énoncé, le MétaContenant 'MC' *existe* par ce premier Lien qu'il a avec lui-même.

Le rôle d'une RelationDeContenance est d'identifier et de représenter par ses LienDeContenance les Contents de son ContenantParent. Donc, ce premier Lien 25, le **Lien '1'**, identifie et représente le MétaContenant 'MC' (figure 30B). Le MétaContenant 'MC' est le contenant de l'ensemble des informations (des Liens) nécessaires au fonctionnement d'un Système construit à partir de ce Noyau de quatorze Liens. Donc le MétaContenant 'MC', autrement dit le Lien '1', *identifie et représente* un tel Système; il contient tout dont lui-même.

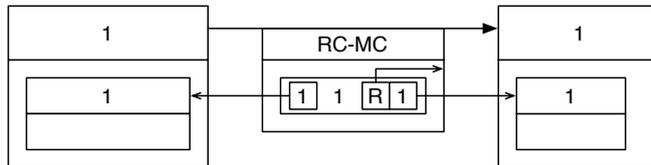


Fig. 30B : le premier Lien identifie et représente le MétaContenant 'MC'

Cependant, l'information identifiant la RelationDeContenance 'RC-MC' et spécifiant quel est son ContenantParent et son ContenantEnfant n'est pas explicitement décrite, le Noyau n'est pas encore totalement auto-décrit. Il faudra encore au moins treize Liens pour exprimer l'information manquante.

6.5 Le Contenant des Relations

Pour exprimer l'information manquante (la RelationDeContenance 'RC-MC'), il faut d'abord définir un Contenant des Relations (figure 31), appelé le **ContenantDesRelations 'CR'**.

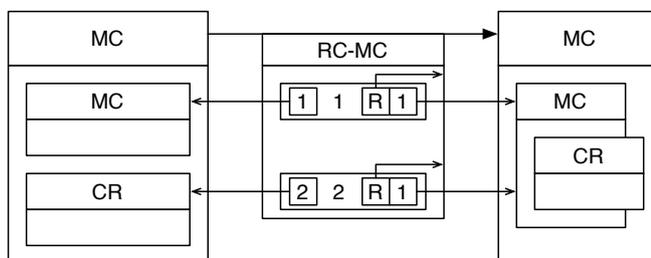


Fig. 31 : ajout du ContenantDesRelations 'CR' au MétaContenant 'MC'

Le **Lien '2'** identifie et représente le ContenantDesRelations 'CR'. Le Lien '2' relie le ContenantDesRelations 'CR' à son Contenant, c'est à dire le MétaContenant 'MC'.

Dans ce Noyau, chaque Contenant doit être défini par au moins une RelationParent : une RelationDeContenance. Le ContenantDesRelations 'CR' a donc besoin d'une RelationParent, la RelationDeContenance 'RC-CR' (figure 32A, p. 28) pour :

- *identifier et représenter* par ses LienDeContenance (Liens '3' et '4') les Contents ('RC-MC' et 'RC-CR') du ContenantDesRelations 'CR',
- *relier* ces Contents ('RC-MC' et 'RC-CR') à leur Contenant (ContenantDesRelations 'CR'), ces Contents étant indiqués par la référence P et leur Contenant indiqué par la référence E des Liens '3' et '4'.

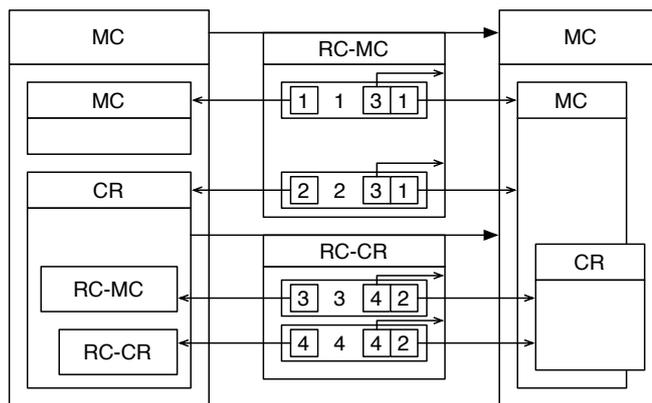


Fig. 32A : ajout de la RelationDeContenance 'RC-CR' pour identifier les deux premières Relations du Noyau ('RC-MC' et 'RC-CR')

Les Contents du ContenantDesRelations 'CR' sont les Relations utilisées pour créer ce Noyau de quatorze Liens. Pour l'instant, le Noyau utilise deux Relations : la RelationDeContenance 'RC-MC' (identifiée par le *Lien* '3') et la RelationDeContenance 'RC-CR' (identifiée par le *Lien* '4').

Les quatre champs des Liens '1', '2', '3' et '4' sont complètement renseignés et tous les éléments (Contenants, Contents et Relations) de la figure 32A sont identifiés et représentés par l'identifiant id d'un LienDeContenance d'une des deux Relations (figure 32B).

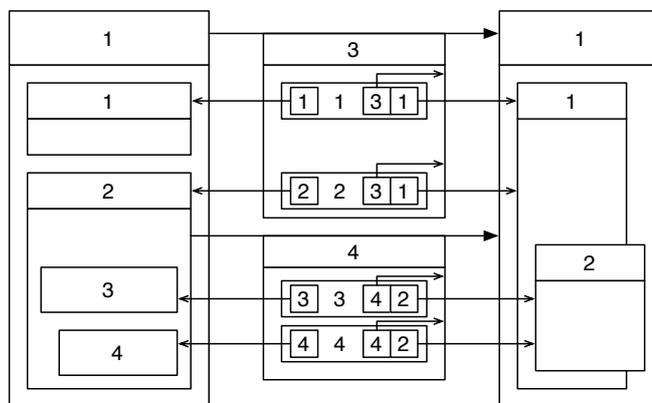


Fig. 32B : le Noyau n'est pas encore auto-décrit, il manque le sens des deux Relations

Pourtant le Noyau n'est pas encore totalement auto-décrit, il reste de l'information qui n'est pas explicitement décrite : le ContenantParent et le ContenantEnfant, désignant le *sens* des RelationDeContenance 'RC-MC' et 'RC-CR', est encore implicite. Il faudra encore au moins dix Liens pour exprimer l'information manquante.

6.6 Modélisation des Relations

Il manque au moins deux RelationParents pour définir les propriétés désignant le *sens* de chaque Relation contenue par le ContenantDesRelations 'CR' (figure 33) : la Relation 'R-CP' (identifiée par le **Lien '5'**) pour désigner le ContenantParent de chaque Relation, et la Relation 'R-CE' (identifiée par le **Lien '6'**) pour désigner le ContenantEnfant de chaque Relation.

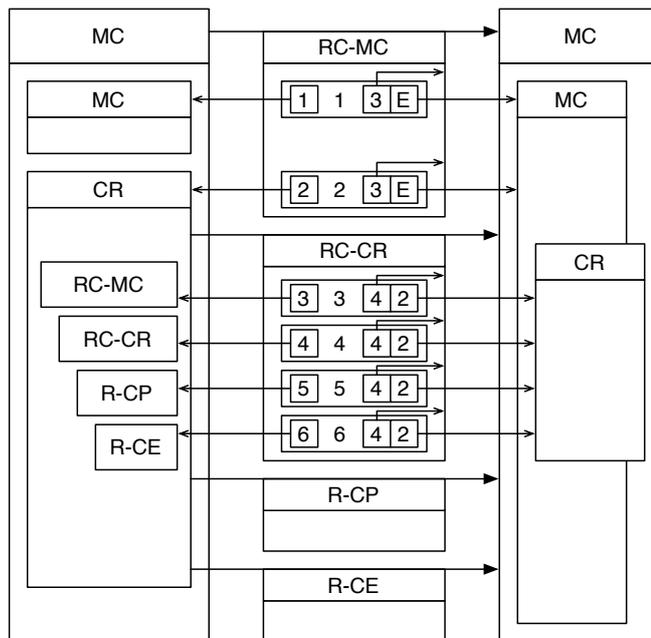


Fig. 33 : modélisation du ContenantDesRelations 'CR' par les Relations 'R-CP' et 'R-CE'

La figure 34 (p. 30) illustre les huit Liens nécessaires pour désigner le ContenantParent (Liens '7' à '10') et le ContenantEnfant (Liens '11' à '14') de chacune des quatre Relations. Ainsi, les quatre champs des quatorze Liens sont complètement renseignés.

Les Liens qui désignent le ContenantParent des Relations :

- Le **Lien '7'** désigne le MétaContenant 'MC' (E '1') comme étant le ContenantParent de la Relation-DeContenance 'RC-MC' (P '3'),
- le **Lien '8'** désigne le ContenantDesRelations 'CR' (E '2') comme étant le ContenantParent de la Relation-DeContenance 'RC-CR' (P '4'),
- le **Lien '9'** désigne le ContenantDesRelations 'CR' (E '2') comme étant le ContenantParent de la Relation 'R-CP' (P '5'),
- le **Lien '10'** désigne le ContenantDesRelations 'CR' (E '2') comme étant le ContenantParent de la Relation 'R-CE' (P '6'),

Les Liens qui désignent le ContenantEnfant des Relations :

- le **Lien '11'** désigne le MétaContenant 'MC' (E '1') comme étant le ContenantEnfant de la Relation-DeContenance 'RC-MC' (P '3'),
- le **Lien '12'** désigne le MétaContenant 'MC' (E '1') comme étant le ContenantEnfant de la Relation-DeContenance 'RC-CR' (P '4'),
- le **Lien '13'** désigne le MétaContenant 'MC' (E '1') comme étant le ContenantEnfant de la Relation 'R-CP' (P '5'),
- le **Lien '14'** désigne le MétaContenant 'MC' (E '1') comme étant le ContenantEnfant de la Relation 'R-CE' (P '6').

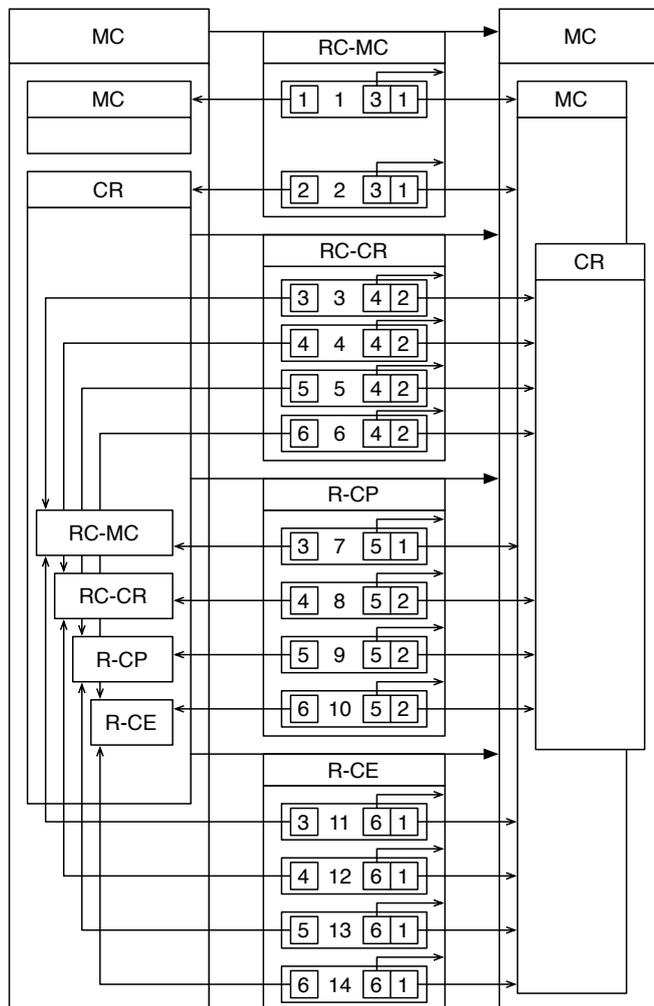


Fig. 34 : le Noyau de quatorze Liens au complet

6.7 Exclusivement fait de Relations

Les deux Contenants, 'MC' et 'CR', sont identifiés par les LienDeContenance '1' et '2' de la RelationDeContenance 'RC-MC', et les *quatre Relations* 'RC-MC', 'RC-CR', 'R-CP' et 'R-CE' sont identifiées par les LienDeContenance '3', '4', '5' et '6' de la RelationDeContenance 'RC-CR'. Les Liens '7' à '14' servent à décrire les ContenantParent et ContenantEnfant des quatre Relations.

La figure 35 remplace chacun des éléments (Contenants, Contenus et Relations) du Noyau (figure 34, p. 30) par l'identifiant id du LienDeContenance qui les identifie et les représente. On constate que tous les Contenants et leurs Contenus sont identifiés et représentés par les Liens d'une RelationDeContenance. La notation schématique des Contenants, de leurs Contenus et des Relations sert à faciliter l'expression d'une modélisation par une représentation Contenant-Relation, mais en fait, tout est Lien.

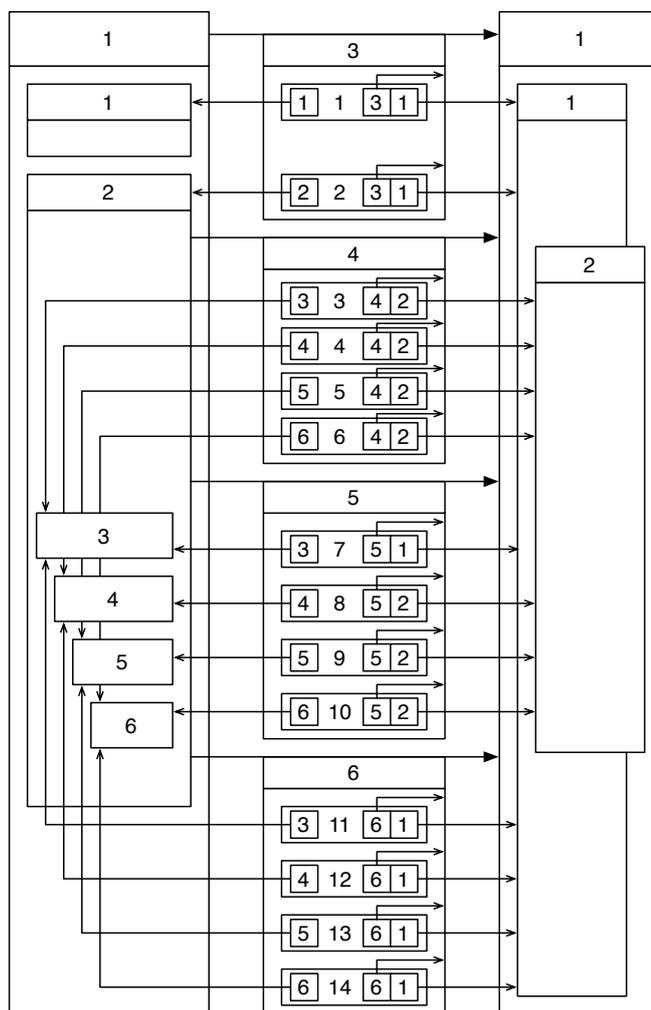


Fig. 35 : tous les éléments sont représentés par des Liens

Le Lien indiqué par la référence R de chacun des Liens est un Lien qui identifie et représente une Relation, ainsi l'ensemble des Liens est organisé en Relations. Un Noyau (ou le Système qu'il permet d'amorcer) est exclusivement constitué par une pluralité de Relations.

La figure 36 illustre le Noyau de quatorze Liens sans la représentation schématique Contenant-Relation. Finalement tout est décrit à l'aide de Liens interconnectés et *reliés*. Ce Noyau élémentaire de quatorze Liens possède toutes les caractéristiques décrites pour amorcer un Système, il est *auto-décrit*, *réflexif*, *autoporteur* et *identifié* par un Lien.

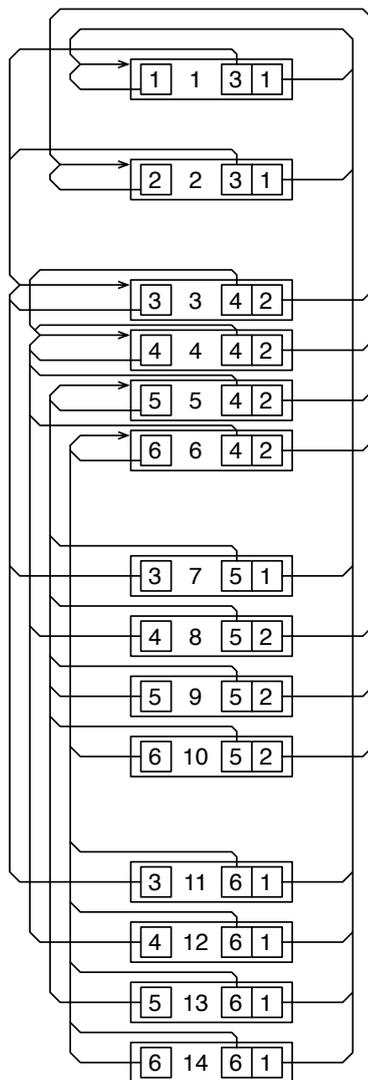


Fig. 36 : Noyau de quatorze Liens sans la notation Contenant-Relation

6.8 Lecture des quatorze Liens

Les quatorze Liens de la figure 34 (p. 30) se lisent :

ContenuParent P	ContenantEnfant E	Lien id	Relation R
le MétaContenant MC	est un Contenu du MétaContenant MC	par le Lien 1	de la RelationDeContenance RC-MC
le ContenantDesRelations CR	est un Contenu du MétaContenant MC	par le Lien 2	de la RelationDeContenance RC-MC
la RelationDeContenance RC-MC	est un Contenu du ContenantDesRelations CR	par le Lien 3	de la RelationDeContenance RC-CR
la RelationDeContenance RC-CR	est un Contenu du ContenantDesRelations CR	par le Lien 4	de la RelationDeContenance RC-CR
la Relation R-CP	est un Contenu du ContenantDesRelations CR	par le Lien 5	de la RelationDeContenance RC-CR
la Relation R-CE	est un Contenu du ContenantDesRelations CR	par le Lien 6	de la RelationDeContenance RC-CR
la RelationDeContenance RC-MC	est associée au ContenantParent MC	par le Lien 7	de la Relation R-CP
la RelationDeContenance RC-CR	est associée au ContenantParent RC	par le Lien 8	de la Relation R-CP
la Relation R-CP	est associée au ContenantParent RC	par le Lien 9	de la Relation R-CP
la Relation R-CE	est associée au ContenantParent RC	par le Lien 10	de la Relation R-CP
la RelationDeContenance RC-MC	est associée au ContenantEnfant MC	par le Lien 11	de la Relation R-CE
la RelationDeContenance RC-CR	est associée au ContenantEnfant MC	par le Lien 12	de la Relation R-CE
la Relation R-CP	est associée au ContenantEnfant MC	par le Lien 13	de la Relation R-CE
la Relation R-CE	est associée au ContenantEnfant MC	par le Lien 14	de la Relation R-CE

Tableau 2 : lecture des quatorze Liens du Noyau

Le Lien '1' identifie le Contenant des Contenants, il *identifie le Système* créé à partir de ce Noyau.

6.9 Réflexivité

La Relation 'RC-CR' (figures 34-35, p. 30-31), identifiée par le Lien '4', est particulière, elle représente la *réflexivité* du Noyau.

La référence R de chaque Lien indique la Relation qui le contient, ainsi la référence R des Liens '11' à '14' indique la Relation '6' dont la référence R indique la Relation '4' dont la référence R indique la Relation '4', c'est à dire elle-même. Il en va de même pour les autres Liens, la référence R des Liens '7' à '10' indique la Relation '5' dont la référence R indique la Relation '4' dont la référence R indique la Relation '4' elle-même. La Relation 'RC-CR' est *contenue par elle-même*, de sorte que la référence R indiquant sa Relation correspond à son propre identifiant id ('4').

7 Enrichissement du Noyau

La difficulté est de créer le premier marteau.
Il est plus facile de fabriquer un marteau à l'aide d'un autre marteau.

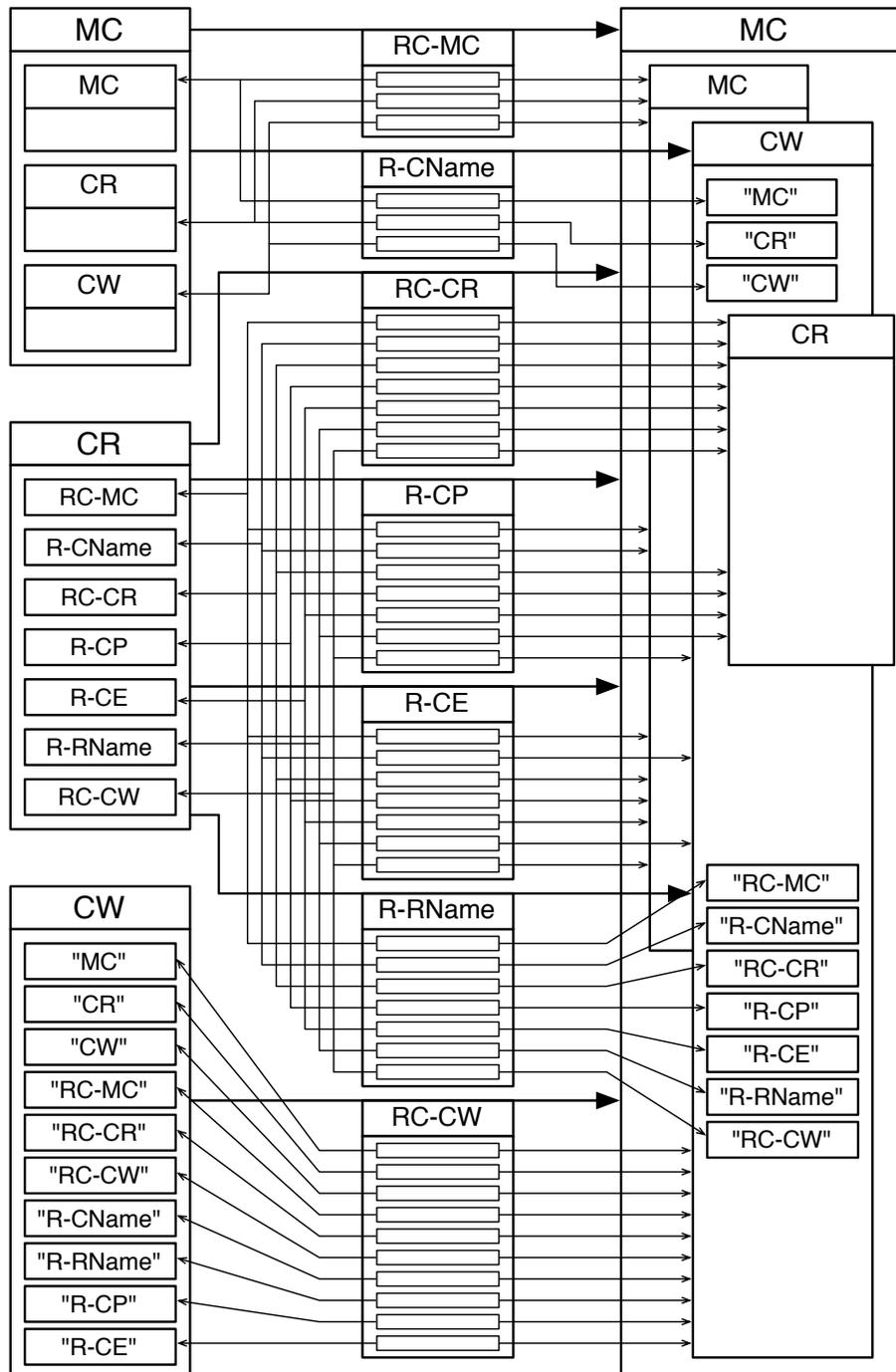


Fig. 37 : un Noyau de quarante-quatre Liens

7.1 Un Noyau de quarante-quatre Liens

La figure 37 (p. 34) illustre un Noyau de quarante-quatre Liens. Il s'agit du Noyau de quatorze Liens *enrichi* d'un nouveau Contenant 'CW'. Les Contenus de 'CW' sont des mots, ils sont identifiés et représentés par les LienDeContenance de la RelationDeContenance 'RC-CW', et ils sont utilisés pour nommer les Contenants et les Relations du Noyau.

Pour permettre une meilleure lecture, la représentation schématique des Contenants est répliquée à gauche et à droite, de façon à placer toutes les Relations au centre de la figure et dans le même sens, avec les ContenantParents à gauche et les ContenantEnfants à droite. En comptabilisant les Liens des sept Relations de ce Noyau, on obtient la somme de quarante-quatre Liens.

Le MétaContenant 'MC' est défini par une Relation supplémentaire : la Relation 'R-CName', qui permet d'attribuer un nom aux Contenants du Noyau. Le ContenantDesRelations 'CR' est défini par une Relation supplémentaire : la Relation 'R-RName', qui permet d'attribuer un nom aux Relations du Noyau. Le Contenant des mots, 'CW', est défini par la RelationDeContenance 'RC-CW', il contient une énumération des mots utilisés par le Noyau.

Une implémentation particulière pourrait définir le Contenant 'CW' par une Relation 'caractères' sur le Contenant 'Unicode' dont les Contenus seraient identifiés par les LienDeContenance de la RelationDeContenance 'RC-Unicode', lesquels LienDeContenance verraient leur id correspondre aux valeurs Unicode. Chaque mot serait ainsi uniquement identifié par un ensemble de Liens sur des caractères. Il y a d'autres implémentations possibles, mais ce n'est pas le propos de cet article.

7.2 Création d'un Modèle A-B

La figure 38 (p. 36) illustre une représentation épurée du Noyau de quarante-quatre Liens. La notation utilisée se lit 'Nom élément / id élément'. 'CW / 3' signifie par exemple que le Contenant 'CW' est identifié par le Lien '3'. Cinq mots, identifiés par les Liens '45' à '49', sont déjà ajoutés au Contenant 'CW' afin de nommer les Contenants et les Relations utiles à la création d'un Modèle A-B, illustré par la figure 39 (p. 36).

La figure 40 (p. 37) illustre une table des Liens possible permettant d'archiver les Liens utiles à la création du Modèle A-B :

- la colonne 'P' représente la référence P du Lien qui indique l'identifiant id du ContenuParent,
- la colonne 'id' représente l'identifiant id du Lien et représente ses attributs temporels,
- la colonne 'R' représente la référence R du Lien qui indique l'identifiant id de la Relation qui contient le Lien,
- la colonne 'E' représente la référence E du Lien qui indique l'identifiant id du ContenuEnfant.

Un curseur temporel 28 simule l'avancement du temps, de manière à pouvoir retracer, Lien par Lien, la séquence de création du Modèle A-B.

La figure 41 (p. 37) illustre l'ensemble des Liens ajoutés de manière séquentielle à la table (les Liens '50' à '73') une fois la création du Modèle A-B terminée. La figure 42 (p. 38) illustre ces mêmes Liens, mais ajoutés dans la représentation épurée du Noyau de quarante-quatre Liens.

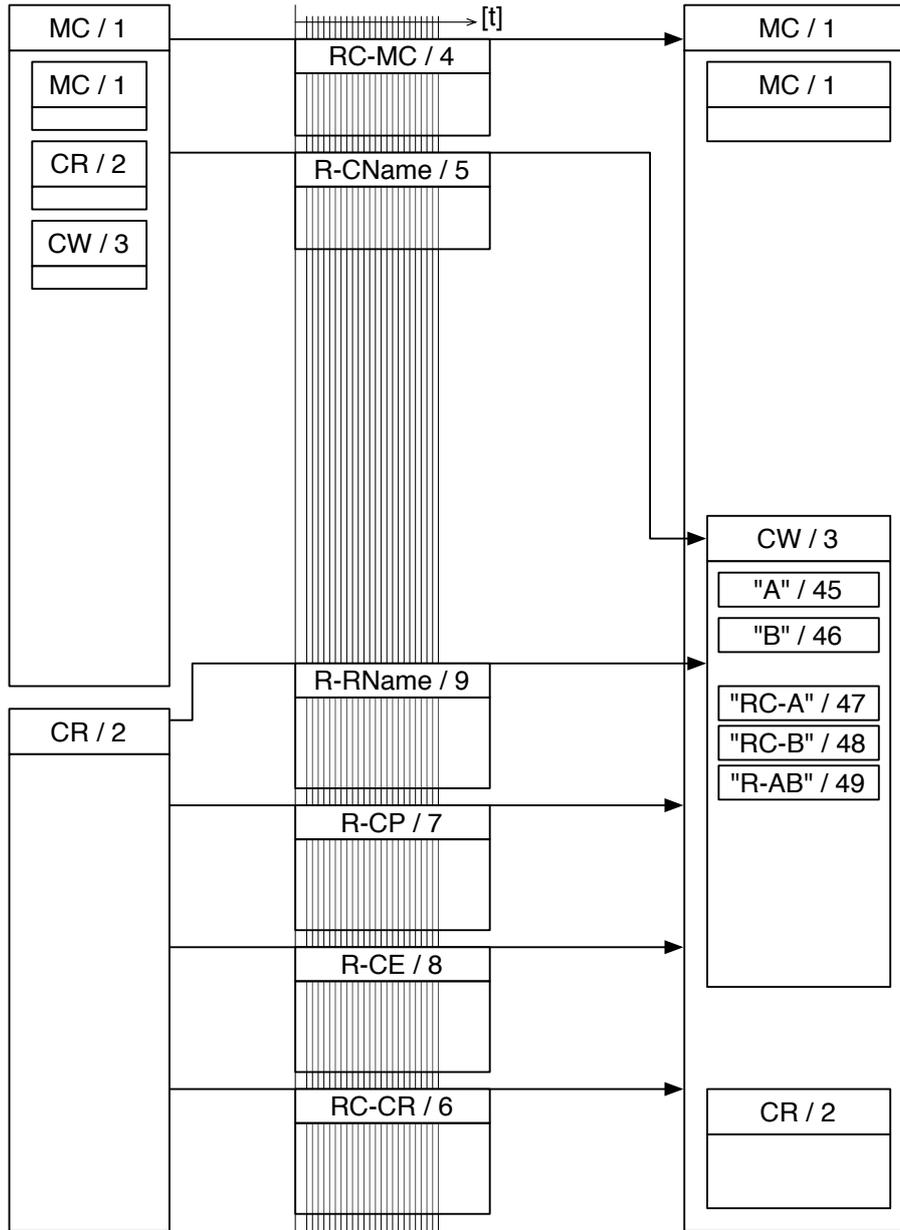


Fig. 38 : Noyau de quarante-quatre Liens, avant la création du Modèle A-B présenté à la fig. 39

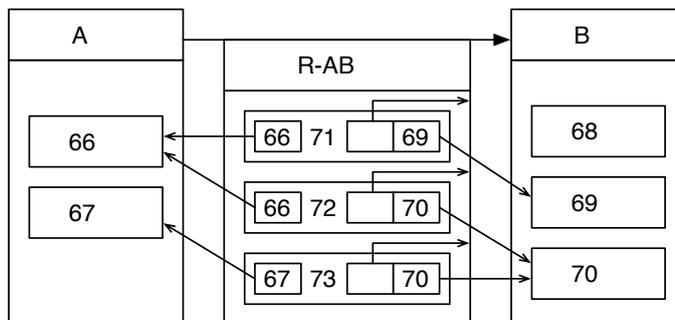


Fig. 39 : le Modèle A-B

P	id	R	E

Fig. 40 : table des Liens, avant la création du Modèle A-B présenté à la fig. 39

P	id	R	E
50	50	4	1
50	51	5	45
52	52	4	1
52	53	5	46
54	54	6	2
54	55	7	50
54	56	8	1
54	57	9	47
58	58	6	2
58	59	7	52
58	60	8	1
58	61	9	48
62	62	6	2
62	63	7	50
62	64	8	52
62	65	9	49
66	66	54	50
67	67	54	50
68	68	58	52
69	69	58	52
70	70	58	52
76	71	62	69
76	72	62	70
77	73	62	70

Fig. 41 : table des Liens, après la création du Modèle A-B présenté à la fig. 39

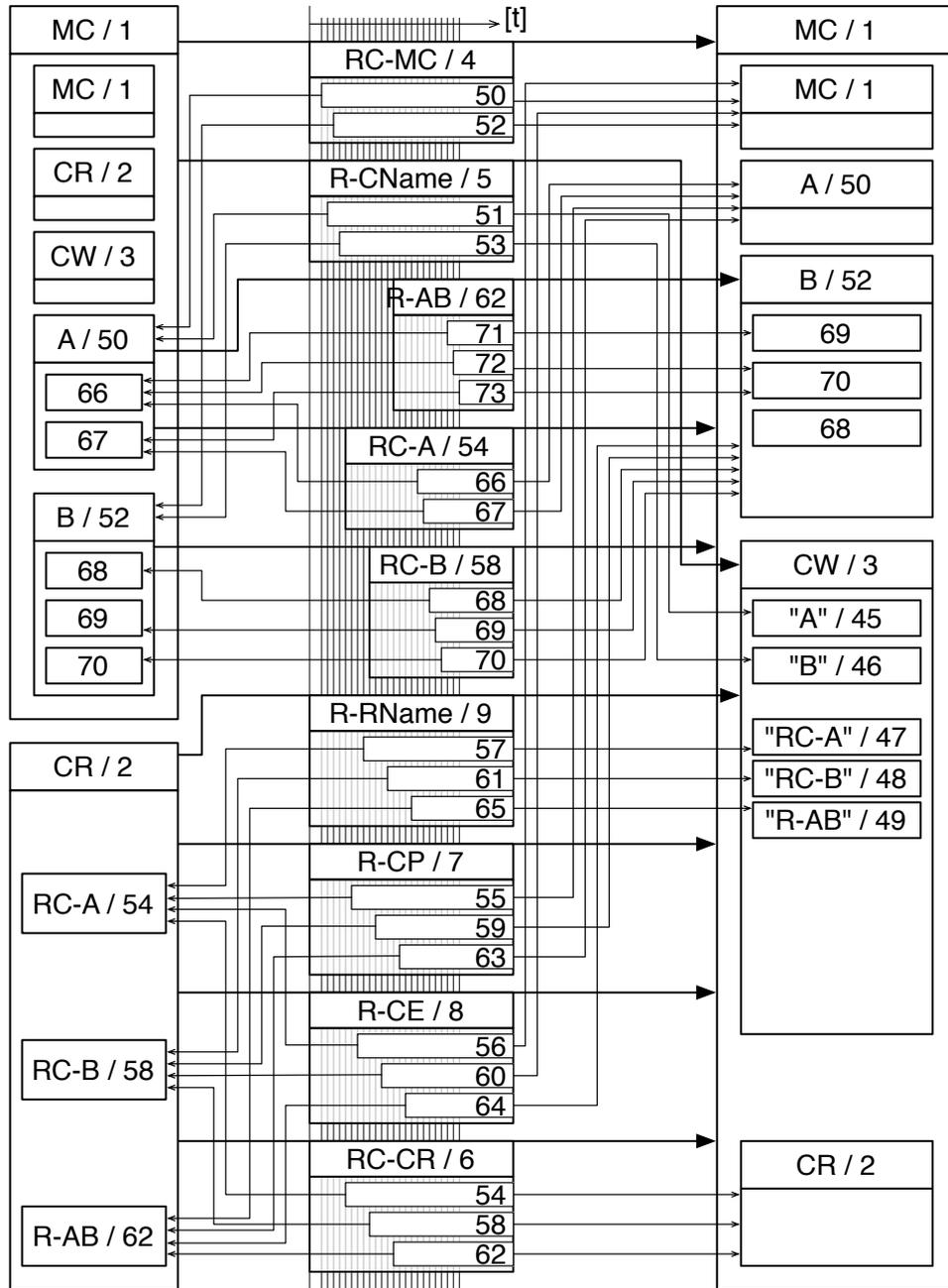


Fig. 42 : Noyau de quarante-quatre Liens, après la création du Modèle A-B présenté à la fig. 39

7.3 Les Liens qui représentent les méta-informations (le contenant)

Les Liens '50' à '65' (figures 41 et 42, p. 37 et 38) représentent les informations de description du Modèle A-B (figure 39, p. 36).

- Un premier Contenant est créé (Lien '50'), il est nommé 'A' (Lien '51').
- Un deuxième Contenant est créé (Lien '52'), il est nommé 'B' (Lien '53').
- Une RelationDeContenance pour le Contenant 'A' est créée (Lien '54'), son ContenantParent est le Contenant 'A' (Lien '55'), son ContenantEnfant est le MétaContenant 'MC' (Lien '56'), elle est nommée 'RC-A' (Lien '57').
- Une RelationDeContenance pour le Contenant 'B' est créée (Lien '58'), son ContenantParent est le Contenant 'B' (Lien '59'), son ContenantEnfant est le MétaContenant 'MC' (Lien '60'), elle est nommée 'RC-B' (Lien '61').
- Une Relation associant le Contenant 'A' au Contenant 'B' est créée (Lien '62'), son ContenantParent est le Contenant 'A' (Lien '63'), son ContenantEnfant est le Contenant 'B' (Lien '64'), elle est nommée 'R-AB' (Lien '65').

L'archivage des méta-informations n'est pas extérieur au procédé, il est également réalisé par des Liens, comme l'archivage des informations elles-mêmes.

7.4 Les Liens qui représentent les informations (le contenu)

Les Liens '66' à '73' (figures 41 et 42 p. 37 et 38) représentent les informations contenues par le Modèle A-B (figure 39, p. 36).

- Deux Contents sont ajoutés au Contenant 'A' (Lien '66' et Lien '67').
- Trois Contents sont ajoutés au Contenant 'B' (Lien '68', Lien '69' et Lien '70').
- Les Contents de 'A' et les Contents de 'B' sont reliés (Lien '71', Lien '72' et Lien '73').

7.5 Règle de création des Contents, Relations et Contenants

La création du Modèle A-B décrit dans les sous-sections précédentes a démontré pratiquement, par la création concrète de Liens, comment créer des Contenants ('A' et 'B'), des Relations ('RC-A', 'RC-B' et 'R-AB') et des Contents (Liens '66' à '73'). Ces trois processus de création sont quasiment identiques, il est possible de les mathématiser par une règle générale.

En règle générale, *créer un nouveau Contenu* 'a' pour un Contenant 'A' consiste à :

- *créer un LienDeContenance* pour la RelationDeContenance 'RC-A' de ce Contenant 'A', de sorte que :
 - ce LienDeContenance va identifier et représenter ce Contenu 'a', sa référence P indique le Contenu 'a' ($P(a) = id(a)$),
 - ce LienDeContenance va relier le Contenu 'a' à son Contenant, le Contenant 'A', sa référence E indique l'identifiant id du Contenant 'A' ($E(a) = id(A)$),
- *créer un Lien pour chacune des autres RelationParents*, de sorte que pour une RelationParent 'R-AB' :

- la référence P du Lien indique le Contenu 'a',
- la référence E du Lien indique un Contenu 'b' du ContenantEnfant 'B' de la RelationParent 'R-AB'.

Pour **créer une nouvelle Relation**, on procède selon la même règle générale de "création de Contenu" ci-dessus, puisqu'une Relation *est* un Contenu, plus particulièrement un Contenu du ContenantDesRelations. En règle générale (et en se référant au Noyau de quatorze Liens illustré par la figure 34, p. 30), créer une nouvelle Relation 'r' pour le ContenantDesRelations 'CR' consiste à :

- *créer un LienDeContenance* pour la RelationDeContenance 'RC-CR' du ContenantDesRelations 'CR', de sorte que :
 - ce LienDeContenance va identifier et représenter la Relation 'r', sa référence P indique la Relation 'r' ($P(r) = id(r)$),
 - ce LienDeContenance va relier la Relation 'r' à son Contenant, le ContenantDesRelations 'CR', sa référence E indique l'identifiant id du ContenantDesRelations 'CR' ($E(r) = id(CR)$),
- *créer un Lien pour chacune des autres RelationParents*, de sorte que pour la RelationParent 'R-CP' :
 - la référence P du Lien indique la Relation 'r',
 - la référence E du Lien indique un Contenu 'C' du ContenantEnfant de la RelationParent 'R-CP'. Le ContenantEnfant de la RelationParent 'R-CP' est le MétaContenant 'MC', et un Contenu 'C' désigne en fait un Contenant, qui est le *ContenantParent* de la Relation 'r'.
 de sorte que pour la RelationParent 'R-CE' :
 - la référence P du Lien indique la Relation 'r',
 - la référence E du Lien indique un Contenu 'C' du ContenantEnfant de la RelationParent 'R-CE'. Le ContenantEnfant de la RelationParent 'R-CE' est le MétaContenant 'MC', et un Contenu 'C' désigne en fait un Contenant, qui est le *ContenantEnfant* de la Relation 'r'.

Pour **créer un nouveau Contenant**, on procède selon la même règle générale de "création de Contenu" ci-dessus, puisqu'un Contenant *est* un Contenu, plus particulièrement un Contenu du MétaContenant 'MC'. En règle générale (et en se référant au Noyau de quatorze Liens illustré par la figure 34, p. 30), créer un nouveau Contenant 'C' pour le MétaContenant 'MC' consiste à :

- *créer un LienDeContenance* pour la RelationDeContenance 'RC-MC' du MétaContenant 'MC', de sorte que :
 - ce LienDeContenance va identifier et représenter le Contenant 'C', sa référence P indique le Contenant 'C' ($P(C) = id(C)$),
 - ce LienDeContenance va relier le Contenant 'C' à son Contenant, le MétaContenant 'MC', sa référence E indique l'identifiant id du MétaContenant 'MC' ($E(C) = id(MC)$),
- *créer un Lien pour chacune des autres RelationParents*. Le MétaContenant 'MC' n'est pour l'instant pas défini par d'autres RelationParents.
- *créer une RelationDeContenance 'RC-C'* pour le Contenant 'C', afin que le Contenant 'C' puisse identifier et représenter ses Contents par les LienDeContenance de sa RelationDeContenance 'RC-C', de sorte que :

- la RelationDeContenance 'RC-C' désigne comme ContenantParent le Contenant 'C',
- la RelationDeContenance 'RC-C' désigne comme ContenantEnfant le Contenant du Contenant 'C', c'est à dire le MétaContenant 'MC'.

Ce dernier point (*créer une RelationDeContenance*) semble constituer une exception à la règle générale. Pourtant, par le jeu de propriétés supplémentaires ajoutées aux Relations (cf. sous-section 7.6), on peut formaliser le comportement de la RelationDeContenance 'RC-MC' en stipulant qu'elle doit être *héritée*, de sorte qu'à la création de chaque nouveau Contenu du MétaContenant 'MC', ce nouveau Contenu (en fait ce nouveau Contenant) hérite d'une RelationDeContenance 'RC'.

En réalité, la règle de création d'un nouveau Contenu/Relation/Contenant est bien sûr plus sophistiquée, la sous-section suivante montre comment certaines propriétés peuvent être précisées.

7.6 Enrichissement des Relations

Le Noyau de quatorze Liens est *autoporteur*, il est capable d'améliorer et d'*enrichir* sa propre définition en créant de nouveaux Contenants et de nouvelles Relations. D'ailleurs, il doit être enrichi de certaines propriétés, pour engager une véritable modélisation de l'information.

On peut enrichir la définition du ContenantDesRelations 'CR' avec de nouvelles RelationParents sur de nouveaux Contenants, afin de préciser les propriétés des Relations, telles que ci-dessous et à titre d'exemples, pour une Relation 'r' :

- les noms de la Relation 'r',
 - par une RelationParent '*nomParent*' sur le Contenant 'Nom de Relation', désignant le nom de la Relation 'r' vu depuis son ContenantParent,
 - par une RelationParent '*nomEnfant*' sur le Contenant 'Nom de Relation', désignant le nom de la Relation 'r' vu depuis son ContenantEnfant,
- le type de la Relation 'r',
 - par une RelationParent '*type*' sur le Contenant 'Type', désignant le type de Relation (cf. sous-section 5.4),
 - par une RelationParent '*obligatoire*' sur un Contenant 'Booléen', désignant si la Relation 'r' *doit* créer au moins un ou plusieurs Liens pour chacun des Contenus de son ContenantParent,
- le comportement de la Relation 'r',
 - par une RelationParent '*héritage*' sur un Contenant 'Booléen', désignant si les ContenuParents de la Relation 'r', au cas où ils représentent également des Contenants, héritent une Relation identique à la Relation 'r',
 - par une RelationParent '*constant*' sur un Contenant 'Booléen', désignant si, lors de la création d'un nouveau ContenuParent, les Liens de la Relation 'r' doivent constamment relier un ou plusieurs ContenuEnfants,
 - par une RelationParent '*par défaut*' sur un Contenant 'Booléen', désignant si, lors de la création d'un nouveau ContenuParent, les Liens de la Relation 'r' relient par défaut un ou plusieurs ContenuEnfants,
 - par une RelationParent '*portée*' sur un Contenant 'Portée', désignant la visibilité de la Relation 'r' (système, privée, publique, ...),

- par une RelationParent '*sémantique*' sur un Contenant 'Sémantique', désignant la sémantique de la Relation 'r' (est un, est composé de, ...),
- par une RelationParent '*parents composés*' sur un Contenant 'Composition', désignant les règles de cardinalité avec d'autres RelationParents du ContenantParent de la Relation 'r',
- par une RelationParent '*enfants composés*' sur un Contenant 'Composition', désignant les règles de cardinalité avec d'autres RelationEnfants du ContenantEnfant de la Relation 'r',
- des informations d'ordre ou de tri,
 - par une RelationParent '*ordre de saisie*' sur un Contenant 'Nombre' pour désigner l'ordre de renseignement de la Relation 'r' parmi les autres RelationParents du ContenantParent de la Relation 'r' lors de la saisie d'un nouveau Contenu,
 - par une RelationParent '*ordre de structure*' sur un Contenant 'Nombre' pour désigner l'ordre de la Relation 'r' parmi les autres RelationParents du ContenantParent de la Relation 'r' pour reconstituer une structure d'information,
 - par une Relation '*ordre des Liens*' associant la Relation 'r' au Contenant 'Nombre' pour désigner l'ordre des ContenuEnfants référencés par les Liens de la Relation 'r',
- etc.

Ces quelques exemples sont mentionnés pour donner une idée des extensions possibles du Noyau. Comment enrichir un Noyau pour lui donner des facultés de modélisation avancées, comment s'organisent les Liens lors de modélisations plus complexes, comment retrouver l'information agencée à l'aide de telles modélisations, etc, sont des questions traitées dans des articles qui demandent un développement trop vaste pour être inclus ici, et qui outrepassent le propos de cet article, centré sur le Noyau.

8 Conclusion et perspectives

Le Noyau de quatorze Liens est la synthèse d'un travail considérable, issu de la volonté de répondre à des questions concrètes et récurrentes, rencontrées dans une longue pratique du génie logiciel, et touchant à des domaines d'activité variés. Ce procédé est en parfaite adéquation avec une réalité connue. Les nouvelles perspectives qu'il libère, répondent à des problèmes bien réels des systèmes d'information actuels. Pour comprendre toute la portée du procédé, il est préférable de l'appréhender dans sa totalité, de le penser en terme de tout et non en termes de parties. On verra plus facilement comment et pourquoi tant d'avantages et d'atouts technologiques *découlent* du procédé. Quelques-uns, qui font l'objet d'articles séparés, sont rapidement résumés ci-après.

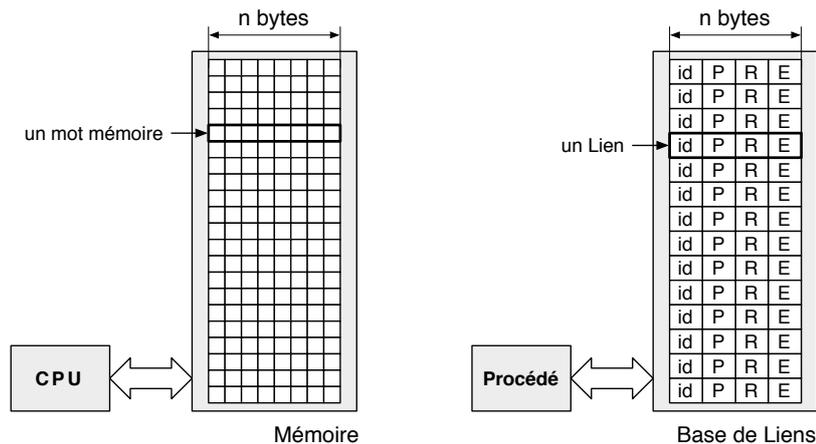


Illustration 3 : unités centrales de traitement

8.1 Attributs temporels

L'information s'exprime dans le mouvement, elle se comprend par le geste. L'information transportée par des mots prend véritablement du sens quand ces mots se succèdent dans une phrase. L'information transportée par une phrase prend tout son sens quand celle-ci est placée dans un discours. Si l'information réside dans les Liens, elle réside tout autant *dans l'évolution des Liens au cours du temps*. C'est la deuxième partie du postulat (cf. 2). Elle n'est pas développée ici parce qu'elle ne concerne pas directement le propos de cet article. Cependant, elle est particulièrement importante : l'information pertinente se trouve dans la transformation de l'information dans le temps, dans son *déroulement*. Sans notion de temps, sans historique, sans trace ou mémoire, il n'y a pas d'informations ni de traitements d'informations possibles.

En ajoutant des *attributs temporels* au cœur du Lien (l'unité élémentaire d'archivage du procédé), on simplifie de manière conséquente la gestion temporelle des informations, des méta-informations, et du Système lui-même. La totalité de l'information peut être stockée dans une simple table (cf. Illustration 3, Base de Liens), elle peut donc être facilement indexée dans le temps (Fig. 41, p. 37). Les attributs temporels du Lien permettent de garder l'historique de toute modification apportée aux informations et aux méta-informations, et aussi de faire des projections de l'information, des simulations. Mais surtout, les attributs temporels transforment le procédé en un *moteur* qui va au-delà d'un procédé de modélisation de l'information tel que présenté dans cet article : le procédé devient un *séquenceur*, capable de "modéliser" des séquences d'informations, et en conditionnant leurs déroulements par des opérations logiques, il est possible de créer des processus, algorithmes ou programmes⁵, et donc d'*ajouter des comportements* aux connaissances représentées par les Liens... Les perspectives sont innombrables.

8.2 Modélisation complexe

Un avantage concurrentiel du procédé est la modélisation d'agencements complexes de structures de données. Bien que l'exemple illustré (cf. 7.2) n'en démontre pas vraiment le potentiel, l'*unicité* des Liens et le fait que le procédé représente la totalité de l'information par des *Relations*, laissent entrevoir une grande souplesse de modélisation. Tout est *identifié* (par l'unicité des Liens) et tout peut être potentiellement *relié* à tout (par les Relations), il n'y a plus de segmentation.

De plus, l'unicité de représentation de l'information par des Liens apporte une solution très simple pour archiver dynamiquement des structures complexes, et les informations qu'elles contiennent, comme des objets par exemple. Selon le procédé, toute modification apportée aux informations et aux méta-informations se traduit simplement par la création, la modification ou éventuellement la suppression de Liens. Le procédé a la capacité intrinsèque de rendre toutes structures de données persistantes. On peut imaginer le procédé comme une solution à la persistance des classes et des objets créés et manipulés par un langage objet : toute modification apportée aux objets et aux classes se traduit simplement par la création, la modification ou éventuellement la suppression de Liens.

Le modèle relié représente un modèle de données accessible, sans grande complexité, où chaque Lien est archivé en tant que Contenu lié par une référence (R) à son Contenant (Relation), et qui est capable de modélisations complexes, au moins aussi complexes que celles autorisées par le modèle objet.

Les possibilités de *modélisation hétérogène, dynamique, évolutive, distribuée*, ou de *marquage sémantique sur plusieurs niveaux*, offrent des perspectives très intéressantes.

8.3 Comportements persistants

Le procédé identifie et représente la totalité de l'information et de la méta-information, l'Hyper Information, uniquement à l'aide de Liens. Les Liens étant fortement interconnectés, il est simple d'intégrer un mécanisme de propagation de *Sélections*.

Une Sélection représente un *flux* qui se transmet de Liens en Liens. Une Sélection est initiée à partir de *Choix* (les Contenus d'un Contenant Source), elle se propage en suivant un chemin tracé par des Relations, et elle s'interrompt en *Impacts* (les Contenus d'un Contenant Cible).

La propagation de Sélections est dite "persistante" parce qu'elle agit comme un courant électrique qui se propage à travers le câblage des Liens. Faire ou défaire un Lien entre deux Contenus agit comme un câble électrique. Le courant passe tant que le câble est connecté, autrement dit tant que les Liens existent ou sont activés. Ainsi les Liens vérifient quasi instantanément la propagation des Multiples Sélections qui agissent et persistent dans les Modèles d'un Système.

On peut effectuer des opérations logiques (OU, ET, etc) sur les Impacts résultants de différentes Sélections initiées depuis divers endroits d'un Modèle, et créer simplement des *filtres persistants*, des *requêtes persistantes*, ou des *règles persistantes*. Une requête persistante maintient dynamiquement le résultat d'une recherche. Si les critères sont modifiés, si les informations changent, ou si la requête est déplacée dans le temps, le résultat est immédiatement mis à jour. On peut faire une comparaison avec des circuits de logique combinatoire, où les sorties réagissent instantanément aux modifications apportées aux entrées. Une autre variante permet d'utiliser les Sélections pour faciliter le déclenchement de mises à jour, et rendre persistant le résultat de *données calculées* ou de *fonctions*.

9 Tables des graphiques

9.1 Les figures

Figure 1a	le Lien	p. 8
Figure 1b	le Lien référence des Liens	p. 8
Figure 2	un Contenu	p. 11
Figure 3	un Contenu est un Lien	p. 11
Figure 4	un Contenant	p. 11
Figure 5	un Contenant et deux Contents	p. 11
Figure 6	un Contenant est un Contenu	p. 11
Figure 7	un Contenant contenu par lui-même	p. 12
Figure 8	un Contenant est un Lien	p. 12
Figure 9	un Lien est identifié	p. 12
Figure 10	la référence R du Lien	p. 12
Figure 11	le Lien relie deux Contents via ses références P et E	p. 12
Figure 12	les différentes notations d'une Relation	p. 12
Figure 13	une Relation est un Contenu	p. 13
Figure 14	une Relation est un Lien	p. 13
Figure 15	une Relation est un Contenant	p. 13
Figure 16	une Relation qui se contient elle-même	p. 13
Figure 17	le ContenantDesRelations	p. 13
Figure 18	une Relation associe un ContenantParent à un ContenantEnfant	p. 14
Figure 19	une Relation associant une Relation à un Contenant	p. 14
Figure 20	le Lien et ses références	p. 14
Figure 21	les RelationParents	p. 15
Figure 22	les RelationEnfants	p. 15
Figure 23	définition récursive des Relations	p. 15
Figure 24	les LienDeContenance de la RelationDeContenance 'RC'	p. 16
Figure 25	un Modèle simple	p. 18
Figure 26A	Relation "de un vers un"	p. 20
Figure 26B	Relation "de un vers plusieurs"	p. 20
Figure 26C	Relation "de plusieurs vers un"	p. 20
Figure 26D	Relation "de plusieurs vers plusieurs"	p. 20
Figure 27	modélisation des Relations 'prénom' et 'nom'	p. 21
Figure 28	tous les éléments du Modèle de la fig. 25 sont identifiés et représentés par des Liens	p. 23
Figure 29	le MétaContenant 'MC'	p. 26
Figure 30A	le premier Lien '1' du Noyau de quatorze Liens	p. 26
Figure 30B	le premier Lien identifie et représente le MétaContenant 'MC'	p. 27
Figure 31	ajout du ContenantDesRelations 'CR' au MétaContenant 'MC'	p. 27
Figure 32A	ajout de la RelationDeContenance 'RC-CR' pour identifier les 2 premières Relations du Noyau ('RC-MC' et 'RC-CR')	p. 28

Figure 32B	le Noyau n'est pas encore auto-décrit, il manque le sens des deux Relations	p. 28
Figure 33	modélisation du ContenantDesRelations 'CR' par les Relations 'R-CP' et 'R-CE'	p. 29
Figure 34	le Noyau de quatorze Liens au complet	p. 30
Figure 35	tous les éléments sont représentés par des Liens	p. 31
Figure 36	Noyau de quatorze Liens sans la notation Contenant-Relation	p. 32
Figure 37	un Noyau de quarante-quatre Liens	p. 34
Figure 38	Noyau de quarante-quatre Liens, <i>avant</i> la création du Modèle A-B présenté à la fig. 39	p. 36
Figure 39	le Modèle A-B	p. 36
Figure 40	table des Liens, <i>avant</i> la création du Modèle A-B présenté à la fig. 39	p. 37
Figure 41	table des Liens, <i>après</i> la création du Modèle A-B présenté à la fig. 39	p. 37
Figure 42	Noyau de quarante-quatre Liens, <i>après</i> la création du Modèle A-B présenté à la fig. 39	p. 38

9.2 Les illustrations

Illustration 1	les couches d'une architecture 3 tiers	p. 4
Illustration 2	câblage du Noyau	p. 25
Illustration 3	unités centrales de traitement	p. 43

9.3 Les tableaux

Tableau 1	les techniques antérieures traitent des données et des références sur des données	p. 6
Tableau 2	lecture des quatorze Liens du Noyau	p. 33

Notes

1. De nombreux aspects du procédé ne seront pas traités dans cet article. Il n'est pas possible de couvrir l'intégralité du potentiel du procédé, mais le Noyau en constitue la pièce fondamentale.
2. Il n'est pas exclu, que pour une implémentation particulière du procédé, certaines références indiquent parfois des identifiants externes. Ceci dans le but de synchroniser le Système avec des données existantes et maintenues par l'extérieur, ou synchroniser des informations maintenues par le Système vers l'extérieur.
3. Le diagramme entité-relation (http://en.wikipedia.org/wiki/Entity_relationship) ne donne pas à ses entités la faculté de se contenir elles-mêmes, ni d'être contenues comme les éléments d'une autre entité, elle ne donne pas à une relation la faculté d'être une entité, ni celle d'être un élément d'une entité.
4. Une intégrité référentielle est un jeu de règles qui empêchent d'effacer ou de modifier par inadvertance des information, si cela peut enclencher des problèmes sur une entité (un Contenant) en relation. Par exemple : on définira qu'un livre a un ou plusieurs auteurs. L'intégrité référentielle interdira l'effacement d'un auteur, tant qu'il existera au moins un livre se référant à cet auteur.
5. Actions, fonctions, routines, programmes, processus, algorithmes, *sont des séquences d'informations conditionnées par des opérations logiques* : machine de Turing (http://fr.wikipedia.org/wiki/Machine_de_Turing).

Références

- [1] Edgar Frank Codd, "[A Relational Model of Data for Large Shared Data Banks](#)". CACM 13, No. 6, June 1970
- [2] Wikipedia, Object database, http://en.wikipedia.org/wiki/Object_database
- [3] Wikipedia, Object-relational mapping, http://en.wikipedia.org/wiki/Object-relational_mapping
- [4] Wikipedia, Object-relational impedance mismatch, http://en.wikipedia.org/wiki/Object-Relational_impedance_mismatch
- [5] Wikipedia, Architecture trois tiers, http://fr.wikipedia.org/wiki/Architecture_trois_tiers
- [6] Douglas Hofstadter, "Gödel Escher Bach, les Brins d'une Guirlande Eternelle". InterEditions, 1985
- [7] "Une chose n'a pas d'existence dans l'absolu, elle n'existe que par rapport à une autre chose", [Gilles Deleuze](#)
- [8] Wikipedia, Associative Model of Data (patent WO00/29980), http://en.wikipedia.org/wiki/Associative_Model_of_Data
- [9] Semantic Web, <http://www.w3.org/2001/sw/> RDF, <http://www.w3.org/RDF/>
- [10] Peter Lucas and Jeff Senn, "Toward the Universal Database: U-forms and the VIA Repository". MAYA Design, 2002. http://www.maya.com/web/what/papers/maya_universal_database.pdf
- [11] Marvin Minsky, "The Society of Mind". ISBN-10 : 0671657135, Simon & Schuster, March 15, 1988

Remerciements

Je remercie mon épouse *Marie-Claire Chappuis*, pour sa patience et sa confiance. Je remercie mon frère *Vincent Chappuis* pour son accompagnement et son soutien dans chaque phase de valorisation de ce procédé, et pour toutes les améliorations apportées à cet article. Je remercie *Thierry Cattel* pour m'avoir lancé dans la rédaction de cet article, pour ses nombreuses relectures et ses judicieux commentaires et conseils.

Table des matières

	Résumé	page 1
	Mots-clés	page 2
	Présentation des sections	page 2
1	Introduction	page 3
1.1	Historique	page 3
1.2	État des lieux	page 3
1.3	Conséquences	page 3
1.4	Segmentation sémantique	page 5
2	Postulat	page 6
2.1	Qu'est-ce qu'une information ?	page 6
2.2	Un nouveau paradigme	page 7
3	Le Lien	page 8
3.1	Structure du Lien	page 8
3.2	Base de Liens	page 9
3.3	Champs supplémentaires du Lien	page 9
4	Vocabulaire	page 10
4.1	Quatre termes principaux	page 10
4.2	Notation Contenant-Relation	page 10
4.3	Définition des termes	page 10
4.4	Autres définitions	page 17
5	Un Modèle simple	page 18
5.1	Le Modèle 'Personnes'	page 18
5.2	Un Contenant est défini par ses RelationParents	page 19
5.3	Création d'un nouveau Contenu	page 19
5.4	Types de Relation	page 19
5.5	Modélisation des Relations	page 21
5.6	Déduction des RelationParents	page 22
5.7	Déduction des RelationEnfants	page 22
5.8	Modélisation des Contenants	page 22

5.9	Tout est Lien	page 23
5.10	Le « modèle relié »	page 23
6	Construction d'un Noyau	page 25
6.1	L'amorçage	page 25
6.2	Le Contenant des Contenants	page 26
6.3	La première Relation	page 26
6.4	Le premier Lien	page 26
6.5	Le Contenant des Relations	page 27
6.6	Modélisation des Relations	page 29
6.7	Exclusivement fait de Relations	page 31
6.8	Lecture des quatorze Liens	page 33
6.9	Réflexivité	page 33
7	Enrichissement du Noyau	page 34
7.1	Un Noyau de quarante-quatre Liens	page 35
7.2	Création d'un Modèle A-B	page 35
7.3	Les Liens qui représentent les méta-informations (le contenant)	page 39
7.4	Les Liens qui représentent les informations (le contenu)	page 39
7.5	Règle de création des Contenus, Relations et Contenants	page 39
7.6	Enrichissement des Relations	page 41
8	Conclusion et perspectives	page 43
8.1	Attributs temporels	page 43
8.2	Modélisation complexe	page 44
8.3	Comportements persistants	page 44
9	Tables des graphiques	page 45
9.1	Les figures	page 45
9.2	Les illustrations	page 46
9.3	Les tableaux	page 27
	Notes	page 47
	Références	page 47
	Remerciements	page 47