



ADELI

La LETTRE n° 37

Octobre 1999



Nous avons lu...

« *Idées nettes sur la logique floue* »

Dans cette rubrique “ Nous avons lu ” chacun d’entre nous est invité à faire part de ses bonnes lectures. Voici une présentation succincte d’un récent ouvrage que nous avons reçu des Presses Polytechniques et Universitaires Romandes. (adresse électronique : www.ppur.epfl.ch)

« Est-il meilleur d’utiliser un modèle certain qui représente de trop loin le réel ou un modèle flou qui le représente mieux ? ». Les progrès des techniques informatiques nous permettent, désormais, d’aborder le traitement automatique de modèles flous qui intègrent la complexité et l’incertitude.

Le titre de ce livre, rédigé par Jelena Godjevac, ingénieur en électricité de l’Université de Belgrade, docteur ès sciences de l’École Polytechnique Fédérale de Lausanne, est suffisamment explicite, quant à la nature de son contenu.

Il n’est, évidemment, pas question de résumer cet ouvrage de 113 pages, accompagné d’un CD-ROM, mais seulement de donner envie, aux curieux, de découvrir cette technique, en l’abordant par ses concepts les plus élémentaires.

La logique floue est une très jeune branche des mathématiques : ses principes ont été énoncés, en 1965, par le professeur Lotfi Zadeh (Berkeley) dans l’article intitulé “ Fuzzy Sets ”.

On a, à l’époque, beaucoup ironisé sur cette appellation, en jouant sur la traduction de l’anglais “ fuzzy ” par le mot “ flou ” qui évoque l’indétermination et, quelquefois, une propension à la dissimulation. Mais, aujourd’hui, lorsqu’on apprend que l’industrie japonaise applique avec succès cette technique pour la régulation de ses robots, on doit considérer cette nouvelle théorie avec plus de discernement.

Cet ouvrage, remarquablement structuré, s’appuie sur un rappel des notions de logique binaire, supposées connues du lecteur. Selon la logique que l’on nous a enseignée, une proposition ne peut prendre que l’une des deux valeurs : vrai ou faux. Un élément appartient ou n’appartient pas à un ensemble ; le critère d’appartenance d’un élément vis-à-vis d’un ensemble ne peut prendre que l’une des deux valeurs, notées 0 et 1. Par exemple, une personne est “ majeure ” si son âge est égal ou supérieur à 18 ans.

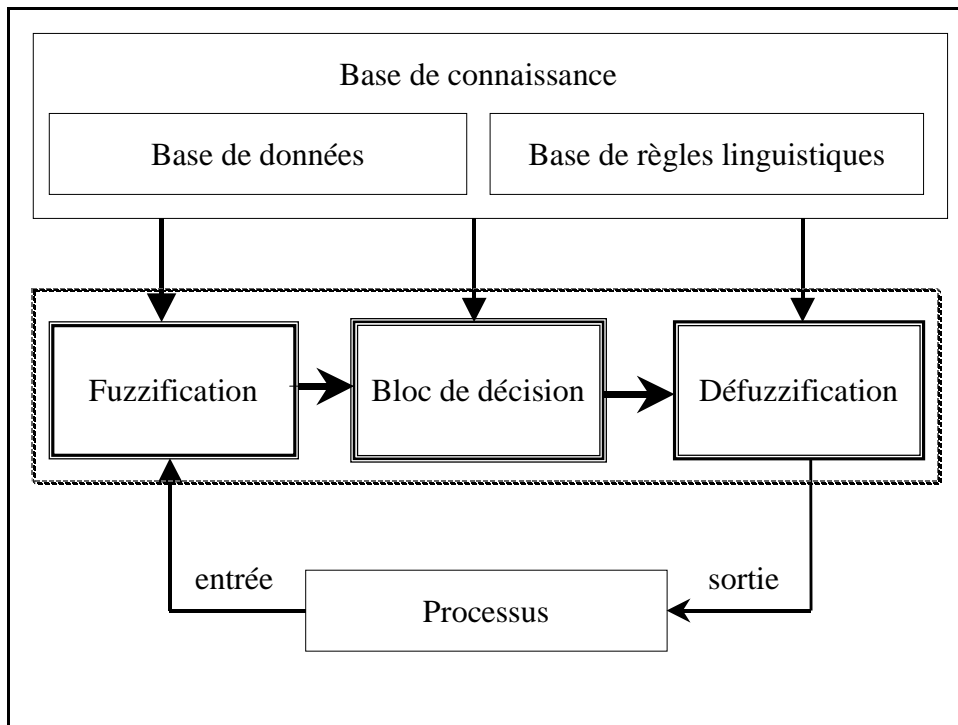
L’ouvrage introduit, ensuite, les nouveaux concepts. En logique floue, un élément est susceptible d’appartenir à un ensemble avec une probabilité comprise entre 0 et 1. Par exemple, si le niveau de maturité d’une personne est lié statistiquement à son âge, on ne peut affirmer avec certitude que telle personne âgée de 20 ans est plus mûre que telle autre âgée de 13 ans.

Ainsi, la variable “ température ambiante ” peut prendre plusieurs valeurs, dites linguistiques : extrêmement froide ; très froide ; froide ; tiède ; chaude ; très chaude ; extrêmement chaude. Tel observateur déclarera l’eau de la piscine “ tiède ” alors qu’un autre la trouvera “ froide ”. On peut associer à ces valeurs linguistiques une probabilité de présence dans un intervalle de température.

Ces développements théoriques suivent une progression pédagogique (seulement freinée par quelques facéties typographiques qui ont écorché certaines notations mathématiques).

La structure d'un système flou de commande d'un processus se compose :

- d'une base de connaissance (règles linguistiques, paramètres des fonctions d'appartenance) ;
- d'un module de " fuzzification " qui transforme les entrées fournies par ce processus en degrés d'appartenance ;
- d'un bloc de décision : moteur d'inférence qui traite les règles linguistiques applicables ;
- d'un module de " défuzzification " qui transforme les résultats flous en commandes du processus.



En d'autres termes,

On saisit des données précises, que l'on replace dans les incertitudes du contexte réel.

Puis, on traite ces informations par des règles extraites du langage courant du genre :

- s'il fait un peu chaud, on baisse le chauffage ;
- s'il fait trop chaud, on ferme le chauffage ;
- si la chaleur est insupportable, on ouvre la fenêtre.
- Ces règles se combinent avec d'autres règles, par exemple sur les difficultés de régler le chauffage, sur les conditions d'ouverture des fenêtres etc.

Enfin, on quantifie la conclusion incertaine obtenue, pour commander une action précise.

La deuxième partie de l'ouvrage comporte des exercices à traiter avec le logiciel LabVIEW, fourni sur CD-ROM. Enfin, la dernière partie présente un petit robot Khepera, muni de deux capteurs et de deux roues motrices, qui, régulé par la logique floue, sait esquiver, en souplesse et en douceur, les obstacles que l'on place sur son trajet.

Qu'en tire un profane ?

Cet ouvrage pédagogique, bien illustré, éclaire les concepts qui permettent d'envisager le traitement des complexités et des incertitudes du monde réel.

Bref, un bon ouvrage d'initiation, avant de savoir s'il faut aller plus loin.

Alain Coulon
Secrétaire d'ADELI
info@adeli.com



Les langages de développement pour le Web

L'arrivée d'XML

Le monde des techniques de développement change. Récemment, le nouveau langage XML est apparu. S'il est utilisé à bon escient par la communauté des développeurs, il peut encore accroître l'interactivité du Web tout en utilisant moins de bande passante¹. Malheureusement, aucun navigateur ne semble supporter XML pour l'instant.

Les différentes techniques de développement utilisées sur le Web

	Techniques de programmation	Techniques de description
sur le serveur	CGI	SQL
sur le client	Java, Javascript	HTML, XML, XHTML, CSS, SMIL

CGI

Le **CGI** ou **Common Gateway Interface** est un programme généralement écrit dans un langage de programmation classique, par exemple le C, qui traite les informations qui lui sont envoyées par le client et génère une réponse, au format **HTML**, correspondant à la demande du client. Si le client veut préciser sa demande, il doit renvoyer une nouvelle demande à laquelle le serveur, qui ne se souviendra sûrement pas de la précédente, répondra à nouveau. D'où une utilisation importante de la bande passante dans un réseau aujourd'hui fort embouteillé. Ce système peut être utilisé pour interroger une base de données via **SQL**.

Java

Il s'agit d'un langage de programmation compilé (développé par Sun) qui, à la différence du **CGI**, sera exécuté sur la machine cliente et pourra donc, une fois le programme et les données transférés, ne plus contacter le serveur. D'où un gain de temps et de bande passante (si le programme est intelligemment développé).

Javascript

Ce langage de programmation (développé par Netscape sous le nom de **Livescript**) n'est pas compilé. Il est intégré dans le contenu des pages **HTML** ou dans des fichiers en .js appelés par celles-ci. Il permet des actions plus limitées que son grand frère **Java**.

HTML

L'**HyperText Markup Language** est un langage de description utilisé dès 1990 par le CERN et toujours utilisé aujourd'hui pour la plupart des pages du Web. Issu de **SGML** (qui est très utilisé dans le monde de l'édition papier), il décrit, à l'aide de balises, la mise en page des documents et n'offre pour seule interactivité que sa fonction hypertexte. Ses spécifications, comme celles de l'**XML** et de l'**XHTML** sont actuellement définies par le W3C (World Wide Web Consortium).

¹ La bande passante d'un réseau ou d'une ligne de communication est le débit supporté par ceux-ci. Plus la bande passante est élevée, plus les données peuvent être transmises rapidement. À l'inverse, plus la bande passante est réduite ou utilisée, plus les données sont transmises lentement.

XML

L'**eXtended Markup Langage** est une généralisation de l'**HTML** qui permet à chacun de créer son propre langage de balisage correspondant à ses besoins. Associé à **Java**, il permet de créer des systèmes de bases de données aussi puissants que l'association des **CGI**, **SQL** et **HTML** en limitant l'utilisation de la bande passante. Les données ne sont transmises qu'une fois et peuvent ensuite être retraitées par la machine du client à la place du serveur. L'**XML** permet de plus de créer un seul document qui pourra être utilisé aussi bien sur un ordinateur de bureau que sur un « organisateur » grâce au traitement du document par des feuilles de style différentes au format **CSS**.

L'**HTML** a récemment été reformulé pour respecter la norme **XML** sous le nom d'**XHTML** ou **eXtended HyperText Markup Langage**.

Un autre langage de balisage, à la norme **XML**, **SMIL**, a été développé par le W3C. Il permet de créer des animations multimédia.

CSS

Les **Cascading Style Sheets** sont les feuilles de style du Web dont le format est défini par le W3C. Elles permettent, à partir d'un seul document au format **HTML**, **XML** ou **XHTML**, d'afficher une présentation différente suivant l'appareil de consultation utilisé par le lecteur. Elles permettent également d'appliquer facilement un même style à une infinité de documents et de changer ce style en ne modifiant qu'un seul fichier (la feuille de style).

Langages propriétaires

Ne sont évoqués ici que les langages reconnus par tous les éditeurs (ou presque). D'autres langages comme **ASP** de Microsoft (de type **CGI**) ou **Flash** de Macromédia (plutôt de type **Java**) existent et sont couramment utilisés mais dépendent d'un éditeur précis.

Certains langages évoqués ne sont pas interprétés de manière identique par tous les logiciels ce qui complique fort le travail des programmeurs. Quelques exemples sont l'**HTML**, **Java** et **Javascript**, qui donnent des résultats différents avec Communicator de Netscape et Internet Explorer de Microsoft. Espérons que le tout nouveau **XML** échappera à ce problème ...

Quelles utilisations ?

Présentations simples

Aujourd'hui, l'**HTML**, simple d'emploi et lisible par tous, reste la meilleure solution. Ne pensez pas utiliser **XHTML** et les **CSS** que lorsqu'un même document doit être lisible sur différents appareils, de l'Organiseur au Macintosh.

Certains sites utilisent même encore de simples fichiers textes... Alors n'alourdissez vos fichiers avec des animations que si cela est vraiment nécessaire.

Présentations animées

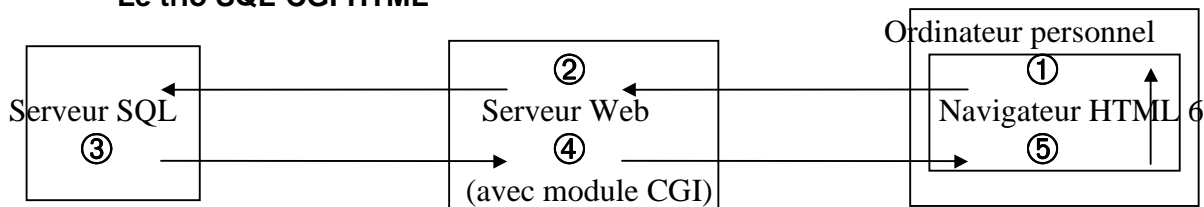
Donc, si c'est vraiment nécessaire, et à part **Java**, **Javascript** et les **GIFs animés**, seules des solutions propriétaires s'offraient à nous jusqu'à aujourd'hui (**Flash**, **RealVideo**, et bien d'autres...). Il est néanmoins préférable d'utiliser des solutions non propriétaires car les utilisateurs se lassent rapidement d'avoir à charger plugin sur plugin !

L'année dernière, grâce au tout nouveau **XML**, **SMIL** est apparu. Il devrait permettre de créer des animations multimédia complexes et lisibles par tous.

Bases de données

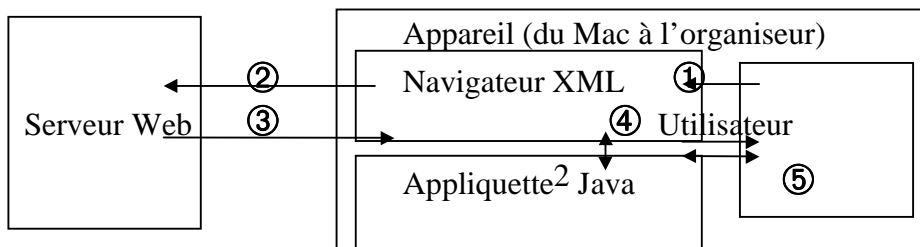
Aujourd'hui, la grande majorité des sites utilise les **CGI** et **SQL** pour permettre l'accès à des bases de données à partir du Web. Ces technologies, très friandes en bande passante et en puissance serveur, seront très certainement supplantées très prochainement par **XML** couplé à **Java** qui est aussi puissant mais n'est pas victime de ces deux défauts. En revanche, **XML** demandera des machines clientes plus puissantes ce qui le rendra inaccessible aux vieux ordinateurs personnels.

Le trio SQL-CGI-HTML



- ① Le navigateur met en forme la demande du client (généralement réponse à un formulaire) qui est ensuite envoyée au serveur Web.
- ② Celui-ci reformule la demande grâce à un module CGI puis l'envoie à la base de données SQL.
- ③ Celle-ci cherche les informations correspondantes dans la base de données et renvoie les informations trouvées au module CGI.
- ④ Celui-ci les met en forme au format HTML puis les envoie au client.
- ⑤ Le navigateur interprète l'HTML et affiche les résultats mis en page.
- ⑥ Si le client souhaite préciser sa demande, tout le circuit doit être recommencé.

Le couple XML-JAVA



- ① L'utilisateur demande l'accès à une base de données
- ② Le navigateur XML demande la base de données correspondante au serveur Web.
- ③ Celui-ci renvoie la base de données au format XML ainsi qu'une applet Java pour la traiter.
- ④ Le navigateur affiche le fichier XML et lance l'applet Java qui peut éventuellement faire, dès ce stade, une sélection dans la base de données.
- ⑤ L'utilisateur peut préciser sa demande grâce à l'applet Java qui cherche les données, non pas en contactant à nouveau le serveur mais en accédant au contenu du fichier XML et en affichant seulement une partie de son contenu via le navigateur.

Inventaire des sources

Livres et articles

Le langage XML de Jon Bosak et Tim Bray dans Pour la Science de Juillet 1999, p.48-54

Sites Web

Le site du W3C, une référence pour les développeurs du Web : www.w3.org

Un service du W3C permettant de vérifier que sa page est bien compatible avec la dernière version d'HTML : validator.w3.org

Noé Lavallée
noe.lavallee@netcourrier.com

² L'applet est la version française de l'applet.



De la vie artificielle vers l'âge de raison

Le comportement des robots nous ouvrirait-il de nouvelles perspectives ?

Lors d'une conférence, intitulée « De l'intelligence artificielle à la vie artificielle », donnée à Paris, le 21 juin 1999, le professeur Hugues Bersini de l'Université libre de Bruxelles, a fasciné son public en présentant les dernières réalisations de son laboratoire de recherche, qui établit des ponts entre la biologie et l'informatique. Par ailleurs, Le Monde a publié, le 6 août 1999, une courte nouvelle de science-fiction intitulée « L'âge des raisons ».

Nous avons exploité cette coïncidence en mélangeant les données de ces deux événements, dans cet article dépourvu de toute prétention scientifique ou philosophique.

La vie artificielle

Le cycle de vie

La vie de tout être suit un cycle inexorable.

Un nouvel être naît de la volonté d'un (ou de plusieurs) concepteur(s).

Puis progressivement, cet être se développe :

- physiquement, en devenant plus grand et plus fort ;
- intellectuellement, en acquérant des connaissances ;
- efficacement, en mettant en pratique les compétences ainsi acquises.

Enfin, cet être vivant, à la suite d'une rupture brutale ou après une lente dégénérescence de ses facultés physiques et intellectuelles, meurt en restituant, à la nature, des composants inertes.

La complexité

La complexification marque le parcours de la « galerie de l'évolution » qui mène des cellules élémentaires aux organismes les plus perfectionnés. Cette complexité naît et prospère d'une répétition, dans le temps et dans l'espace, de mécanismes extrêmement simples. Ainsi, la répétition d'opérations élémentaires débouche sur des phénomènes émergents d'une extraordinaire complexité.

Voilà qui rapproche l'informatique de la biologie.

Grâce à l'informatique, les biologistes acquièrent une perception nouvelle des systèmes qu'ils étudient. Ils ont la possibilité d'effectuer des simulations informatiques afin de tracer le passage de la simplicité à la complexité, par exemple :

- croissance des végétaux ;
- apparition et disparition des espèces animales ;
- comportements collectifs des sociétés d'insectes ;
- phénomènes dynamiques du cerveau.

Le jeu de la vie

Pour illustrer la génération de la complexité, le professeur Bersini rappelle les règles de ce jeu.

Ce jeu consiste à choisir, dans une matrice, de dimensions finies, une cellule de départ et de donner une règle simple de génération des propriétés (par exemple la couleur) des cellules adjacentes (gauche, supérieure, droite, inférieure) et de répéter cette opération sur l'ensemble de la matrice, sans oublier les règles de rebond, sur les bords du tableau.

Ces algorithmes, sont très facilement programmables. Leur automatisation permet de visualiser la cinématique des résultats obtenus. On observe des configurations, totalement imprédictibles ; quelques-unes convergent vers une situation stable alors que d'autres entretiennent des évolutions perpétuelles.

Les robots du professeur Bersini

Puis, le professeur Bersini présente quelques exemples de robots¹ informatiques, conçus et réalisés en laboratoire, qui ont appris à exécuter des opérations complexes.

Un « lézard » marche sur ses quatre pattes.

Un « singe » saute de branche en branche.

Des « combattants » s'affrontent pour s'emparer d'une proie.

Il faut bien noter que ces robots ne se contentent pas de répéter mécaniquement certains mouvements inculqués ; au fur et à mesure de leur vie, ils améliorent leur savoir-faire en capitalisant les leçons de leurs expériences.

Une nouvelle de science-fiction

Ce thème de la vie artificielle est le décor d'une courte nouvelle de la romancière Nancy Kress « L'âge des raisons » publiée dans le quotidien « Le Monde » du 6 août 1999.

Sans reprendre l'ensemble de cette nouvelle, nous nous attacherons aux aspects qui la rapprochent de la communication du professeur Bersini.

Des robots surdoués

Une entreprise de haute technologie a développé des robots conditionnés pour :

recueillir des jetons de différentes dimensions, que l'on déverse dans un enclos garnis d'obstacles (labyrinthes, dénivellations, fossés, etc.) ;

les restituer, en les introduisant par des fentes, correspondant aux formes des jetons.

Chaque robot, sous une enveloppe métallique géométrique simple, est muni :

d'organes sensoriels, pour appréhender l'espace dans lequel il se meut ;

de quatre pattes, pour se déplacer ;

de deux bras articulés, pour attraper les jetons.

Par stades successifs, les robots s'enrichissent de leurs expériences pour acquérir l'équilibre, le mouvement, le contournement d'obstacles, la préhension d'objet.

Ils découvrent, par eux-mêmes, la synergie du travail en équipe, pour combiner leurs efforts, de façon à accroître leur productivité, en utilisant au mieux leurs particularités et leurs aptitudes respectives.

Ils apprennent à anticiper en préparant les tâches auxquelles on va les soumettre.

L'homme d'affaires

L'existence et les performances de ces prototypes ne manquent pas d'attirer, en rangs serrés, les convoitises d'entrepreneurs qui entrevoient de futures affaires mirifiques.

L'enthousiasme d'un de ces concurrents est freiné par des difficultés familiales. Son fils adolescent l'inquiète particulièrement et cette préoccupation entrave ses négociations.

¹ « Robot » mot dérivé d'une racine slave qui signifie « travail », inventé par l'auteur tchèque Karel Čapek.

La chronologie des âges

Considéré, par son entourage, comme un doux contemplatif, ce fils atypique passe le plus clair de son temps à réfléchir sur la succession des âges qui jalonnent l'évolution de l'homme : âge de pierre ... âge du bronze ... moyen âge ... âge des lumières ... âge de l'industrie ... âge de l'espace ... âge de l'information.

En particulier, il s'appesantit sur l'âge des lumières du XVIII^{ème} siècle qui marque une relance de l'activité intellectuelle vers une recherche des causes et une remise en cause de l'autorité.

Quelle étrange préoccupation !

Alors que le commun des mortels, confortablement enfermé dans l'âge de l'information, considéré comme un achèvement, se refuse à envisager tout changement, qui ne pourrait être qu'une récession !

Du grain de sable à la grève

Revenons aux robots que l'on vient de connecter, entre temps, au réseau informationnel. Leur comportement semble se modifier, de façon inexplicable.

Dans un premier temps, ils se contentent de récolter les jetons sans les introduire dans les fentes de restitution, comme ils le faisaient, avec dextérité, quelques jours auparavant.. Puis, peu après, ils s'abstiennent ostensiblement de tout mouvement. Le lancement des jetons qui provoquait, naguère, la belle effervescence attendue, tombe désormais dans une profonde indifférence.

Toute incitation pour animer cette génération de robots se révèle infructueuse.

La raison

Dans une première phase, les robots semblent avoir suivi une progression logique, héritée de nos comportements humains :

Comment faire ce que l'on me demande : attraper des jetons et les glisser dans les fentes ;

Comment le faire mieux et plus vite : augmenter la productivité en améliorant mes performances individuelles et en établissant une synergie avec mes collègues.

En cette fin de millénaire, notre société en est à stade d'évolution.

Mais après avoir atteint leurs résultats, dans une seconde phase, les robots, abreuvés par une multitude de sources d'informations qu'ils prennent le temps d'assimiler, extrapolent leur évolution en se demandant :

Pourquoi restituer les jetons à ceux qui, dans quelques instants, vont les faire pleuvoir, à nouveau, sur notre enclos ?

À quoi bon ramasser les jetons ?

Ainsi,

avant le niveau physique, qui ordonne la séquence des travaux ;

avant le niveau logique, qui choisit le « comment » (les moyens techniques) ;

avant le niveau organisationnel qui détermine le « qui », le « quand » et le « où » ;

avant le niveau conceptuel qui définit le « quoi » ;

il y aurait un niveau préliminaire qui s'interrogerait sur le « pourquoi ».

Les robots, non seulement, ont parcouru, en quelques semaines, le cycle de l'évolution, mais après l'âge de l'information, ils nous ont précédés dans l'âge des raisons.

D'autre part, pour résister à l'oppression, nos robots démontrent qu'une inertie résolue et tranquille est souvent plus efficace qu'une contestation désordonnée et bruyante.

Invitation à la réflexion

Bien sûr, la nouvelle que nous venons d'évoquer n'est qu'une aimable utopie.

Mais, rapprochée des progrès conjoints de la biologie et de l'informatique, tels qu'ils sont illustrés par les robots du professeur Bersini, la chute de cette histoire prend une résonance singulière.

Devons-nous omettre de nous interroger sur la finalité de nos actions ?

Avant de nous lancer brutalement dans des actions de résistance violente, sommes-nous certains d'avoir épuisé les prodigieuses ressources de l'inertie ?

*Alain Coulon
Secrétaire d'ADELI
info@adeli.com*



Commission

Réseau Sémantique Universel

Étude de cas n°1 : routage postal

La meilleure méthode pour mettre au point une méthode étant de la tester sur le plus grand nombre possible de cas concrets, voici le premier exemple soumis à la sagacité des membres de la Commission RSU.

Note : le début de modélisation présenté à la suite de l'énoncé a été fait, non pas par un des membres de la Commission, mais par un étudiant.

Énoncé du problème

Une entreprise de vente par correspondance envoie régulièrement des documents à ses clients. Pour ce faire, elle effectue pour chaque envoi un regroupement en liasses et sacs postaux en fonction des adresses des clients, par département, code postal et quartier, qui économise du travail à la Poste, et lui permet d'obtenir une remise. Ces remises représentant plusieurs millions d'euros chaque année, ce regroupement est donc une activité rentable.

Définitions

Envoi

Un envoi consiste à faire parvenir à certains clients un pli. Tous les plis d'un envoi sont identiques. Un envoi ne concerne généralement qu'un sous-ensemble des clients ; plusieurs envois peuvent être en cours à un moment donné ; un client peut être concerné par plusieurs envois.

Pli

Le pli est le document envoyé à un client. Il peut s'agir d'une simple carte postale pesant 5g, d'une lettre à 20g, d'un prospectus de 100g, d'un catalogue promotionnel de 200g ou d'un catalogue saisonnier pouvant peser plus de 2kg, tous les intermédiaires étant possibles.

Liasse

Une liasse est un regroupement physique de plis, généralement quelques dizaines, mais il peut arriver qu'on obtienne en fin de traitement une liasse d'un seul pli.

Sac

Les liasses ne sont pas transportées telles quelles, mais regroupées en sacs postaux. Il peut arriver qu'un sac ne contienne qu'une seule liasse.

Département

Le département est effectivement la division administrative classique (96 métropolitains + outremer). Pour les clients étrangers, chaque pays est traité comme un département.

Code postal

Il s'agit, en théorie, du code postal utilisé pour le courrier. Cependant, compte tenu de certains codes spéciaux (cedex...) et de communes à double code postal (Lille, Dunkerque...), une correspondance est faite entre le code postal exact du client et un code postal logique.

Quartier

Il s'agit, pour un code postal donné, d'un découpage correspondant aux tournées des préposés à la distribution. Tous les envois d'un même département, code postal et quartier sont distribués par une même personne.

État initial du système

Nous n'étudierons ici que la partie du système utilisée postérieurement à la création d'un envoi et rattachement des destinataires de cet envoi.

Les traitements qui nous intéressent pourront donc être activés par un prédicat "Envoi initialisé". Trois étapes successives sont nécessaires.

Étape 1 - Détermination des nombres de plis par liasse et par sac

Le premier traitement consiste à calculer les valeurs suivantes :

Nombre minimal de plis par liasse : on ne peut constituer de liasse avec un nombre inférieur de plis, sauf cas extrêmes décrits plus loin.

Nombre optimal de plis par liasse : c'est le nombre de plis qui sera utilisé aussi souvent que possible.

Nombre maximal de plis par liasse : une liasse ne pourra jamais contenir un nombre supérieur de plis ; il n'y a pas d'exception.

Nombre minimal de plis par sac.

Nombre optimal de plis par sac.

Nombre maximal de plis par sac.

Pour pouvoir obtenir ces valeurs, on dispose au niveau de l'objet Envoi de deux informations :

Le poids exact d'un pli.

La catégorie du pli : selon que le tri postal peut être automatique (lecture optique) ou pas (à cause de la forme ou du poids du pli), ou bien qu'il s'agisse de plis de poids élevé (catalogues), les poids des liasses et sacs peuvent différer.

On dispose également d'une table, dont chaque objet est identifié par un code catégorie et un poids maximal, et contient les données suivantes :

Poids minimal, optimal et maximal d'une liasse.

Poids minimal, optimal et maximal d'un sac.

Éventuellement un nombre minimal et/ou un nombre maximal de plis par liasse.

On calcule le nombre minimal de plis par liasse comme suit : si un nombre minimal de plis est défini dans la table, c'est cette valeur qui est prise. Sinon, on divise le poids minimal d'une liasse par le poids d'un pli, avec arrondi à l'entier supérieur.

On calcule le nombre maximal de plis par liasse comme suit : si un nombre maximal de plis est défini dans la table, c'est cette valeur qui est prise. Sinon, on divise le poids maximal d'une liasse par le poids d'un pli, avec arrondi à l'entier inférieur. Si le résultat obtenu est inférieur au nombre minimal de plis par liasse, il est incrémenté pour égaliser le minimal.

On calcule le nombre optimal de plis par liasse comme suit : on divise le poids optimal d'une liasse par le poids d'un pli, avec arrondi à l'entier inférieur. Si le résultat obtenu est inférieur au nombre minimal de plis par liasse, ou supérieur au nombre maximal, il est incrémenté pour égaliser le minimal ou décrétementé pour égaliser le maximal.

Pour les nombres minimaux, optimaux et maximaux de plis par sac, le calcul est le même, sauf que le calcul est toujours fait à partir des poids.

Étape 2 - Regroupement des envois par type de routage

Le second traitement consiste à affecter à chaque destinataire un code routage, dépendant du nombre de destinataires dans sa zone.

Chaque département fait l'objet d'un traitement séparé. Il n'y a jamais de regroupement de 2 départements ou plus dans un même sac :

Pour chaque quartier, on comptabilise les plis pour savoir si oui ou non il y a de quoi faire au moins une liasse, et si c'est le cas, de quoi faire un sac ou pas.

Les destinataires de chaque quartier pour lequel on peut remplir au moins un sac reçoivent le code routage 1.

Les quartiers, pour lesquels on pouvait faire des liasses mais pas un sac, sont regroupés au niveau du code postal. Et on vérifie si on peut remplir au moins un sac de liasses de quartier. Si oui, les destinataires de ces quartiers reçoivent le code routage 2.

Tous les quartiers restants sont ensuite regroupés au niveau du code postal :

Pour chaque code postal, on comptabilise les plis pour savoir si oui ou non il y a de quoi faire au moins une liasse, et si s'est le cas, de quoi faire un sac ou pas.

Les destinataires de chaque code postal pour lequel on peut remplir au moins un sac reçoivent le code routage 3.

Les codes postaux, pour lesquels on pouvait faire des liasses, mais pas un sac sont regroupés au niveau du département. Et on vérifie si on peut remplir au moins un sac de liasses de codes postaux. Si oui, les destinataires de ces codes postaux reçoivent le code routage 4.

Tous les autres destinataires reçoivent le code routage 5.

Pour résumer au niveau des sacs postaux :

1 = sac dont tout le contenu est destiné au même quartier. Il sera transmis directement au responsable de ce quartier.

2 = sac dont tout le contenu est destiné au même code postal, contenant des liasses de plis propres à un seul quartier chacune. Il sera transmis directement au bureau de poste principal, qui n'aura qu'à retransmettre les liasses aux divers quartiers.

3 = sac dont tout le contenu est destiné à un seul code postal. Il sera transmis directement au bureau de poste principal, qui déliassera les plis pour les retransmettre aux divers quartiers.

4 = sac départemental contenant des liasses de plusieurs codes postaux, mais chacune propre à un seul code postal. Il sera transmis au centre de tri du département, qui répartira les liasses vers les bureaux de poste concernés, où les plis seront déliassés.

5 = sac départemental contenant des liasses de plis pouvant avoir des destinations variées. Il sera transmis au centre de tri du département, qui déliassera les plis pour les retransmettre vers les différents bureaux de poste.

Note : si un code postal n'a qu'un seul quartier, celui-ci est assimilé au code postal et ne pourra donc pas recevoir de code routage 1 ou 2.

Étape 3 - Regroupement des plis en liasses et sacs

Il reste maintenant à créer logiquement les liasses et les sacs. On va donc attribuer à chaque pli un numéro de liasse, liasse elle-même recevra un numéro de sac. Cette numérotation est indépendante pour chaque département, c'est-à-dire qu'on reprend la numérotation des sacs et liasses à 1 à chaque fois. Mais la numérotation des liasses ne reprend pas à 1 lors d'un changement de sac, ni lors d'un changement de code routage.

On va donc traiter d'abord tous les plis ayant le code routage 1, dans l'ordre des codes postaux, et dans celui des quartiers pour un même code postal. On fera de même successivement avec les codes routage 2 à 5.

Attention : il y a forcément changement de liasse :

en routage 1 et 2, lors d'un changement de quartier ;

en routage 3 et 4, lors d'un changement de code postal.

Il y a forcément changement de sac :

en routage 1, lors d'un changement de quartier ;

en routage 2 et 3, lors d'un changement de code postal.

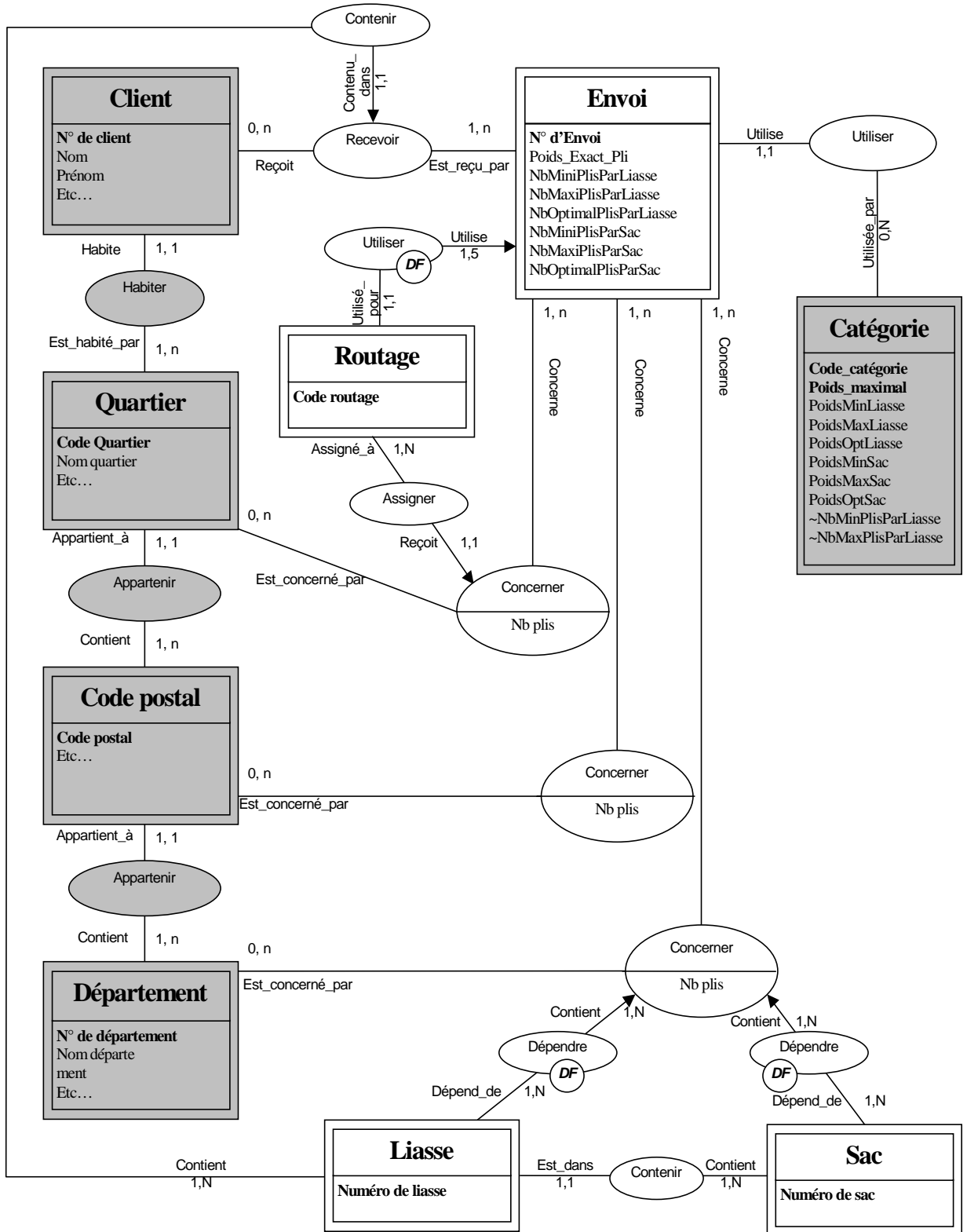
En fin de département, ou lors d'un changement de sac et/ou de liasse pour l'une des causes ci-dessus, le nombre de plis restants ne correspond pas nécessairement exactement à une liasse optimale. Dans ce cas, il faut répartir le manque ou l'excès sur les 2 dernières liasses.

Par contre, le défaut ou l'excès au niveau des sacs sera intégralement supporté par le dernier sac. Un sac est considéré comme rempli de façon optimale lorsque l'ajout d'une liasse ferait dépasser le nombre optimal de plis de ce sac.

Début de modélisation

Dans les schémas présentés ci-après, les objets et relations non spécifiques à l'application concernée par l'énoncé (donc en fait un sous-ensemble de l'application complète de routage, puisque nous n'avons pas pris en compte la création de l'envoi, son exploitation ultérieure et sa suppression, ni la mise à jour de la table des catégories) sont colorés en gris.

Modèle de Données



Le schéma ci-avant montre un premier jet de modélisation des données, qui n'a rien que de très classique, puisque ne contenant aucune "nouveau" RSU. Ce qui est important, c'est que toutes les informations de routage sont dépendantes de l'objet "Envoi".

Pourquoi ? Parce qu'une fois le routage effectué, il suffit de demander la suppression de cet objet pour que tout ce qui s'y rattache soit également supprimé automatiquement.

Pour classique qu'il soit, ce MCD met cependant en lumière une incertitude : comment différencier une relation de relation d'une identification relative, le symbolisme étant actuellement le même ? Ici, on a ajouté un symbole \textcircled{DF} (Dépendance Fonctionnelle) pour lever le doute, mais on peut faire mieux au niveau de la représentation.

Quelques commentaires pour ceux qui n'ont pas une bonne connaissance de Merise ou de la conception en général.

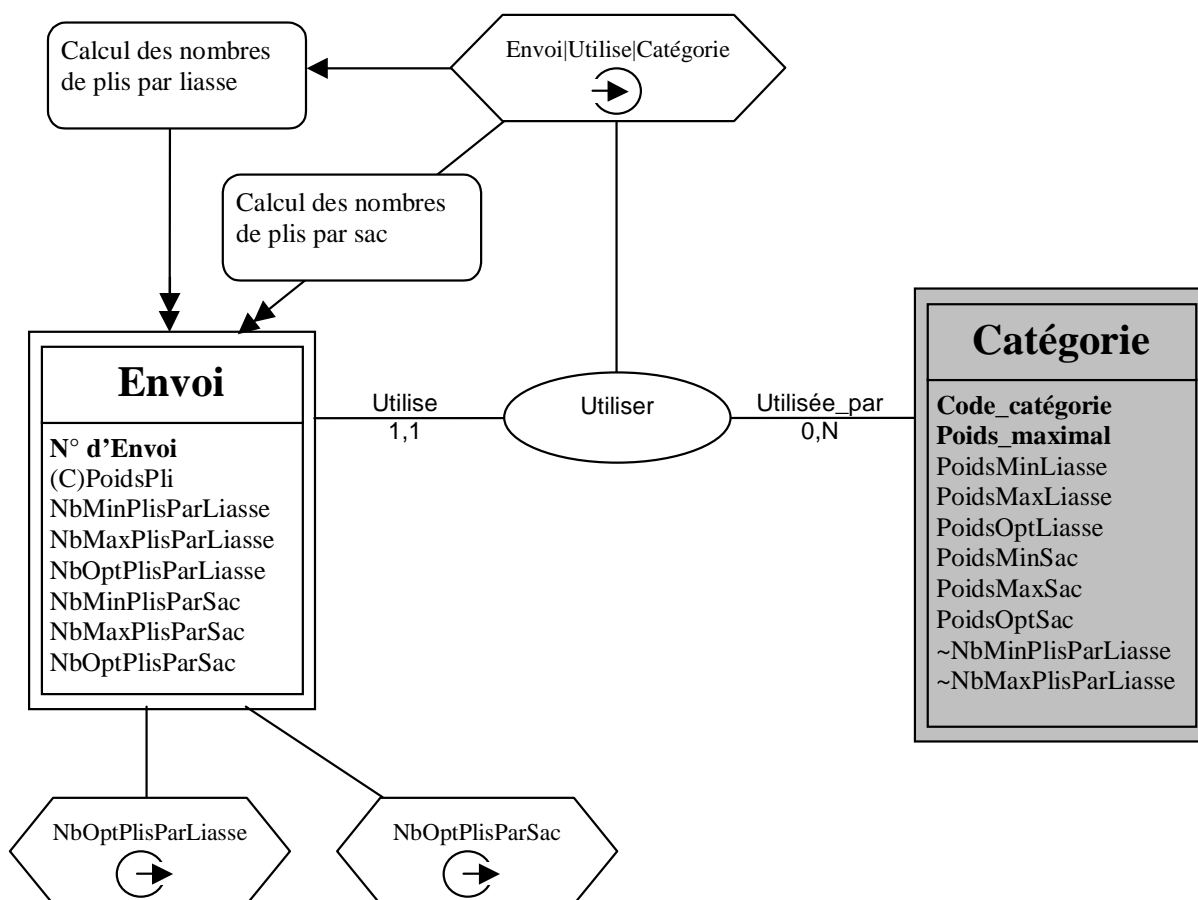
Question : Pourquoi le code routage est-il représenté comme un objet, qui plus est relié à la relation Envoi|concerne|Quartier, et non comme une propriété de la relation Client|reçoit|Envoi qui correspond à la définition du pli ?

Réponse : le code routage est un objet, parce que la propriété correspondante est un identifiant lors du processus de création des liasses et sacs. Il est relié à Envoi|concerne|Quartier, parce que, compte tenu de l'énoncé, tous les clients d'un même quartier ont forcément le même code routage. C'est une application de la quatrième forme normale.

Question : Pourquoi les objets Liasse et Sac sont-ils reliés à Envoi|Concerne|Département ?

Réponse : Parce qu'une liasse et un sac ont un identifiant (leur numéro) spécifique à un département et ne concernent qu'un envoi, ce qui correspond bien à l'identifiant de la relation indiquée.

Modèle général de l'étape 1



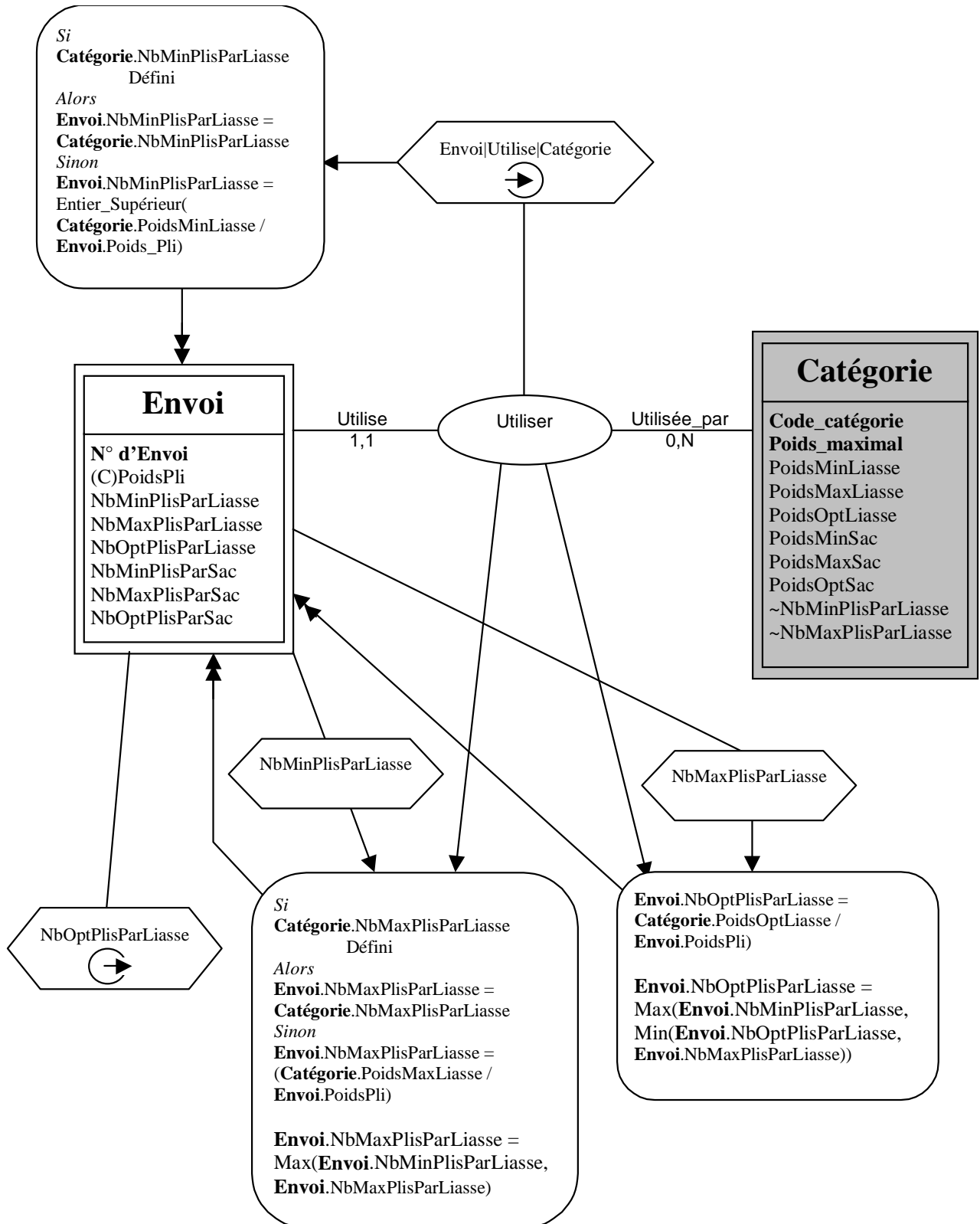
Ce schéma est assez simple. Comme en fait les calculs de nombre de plis par liasse et par sac sont :
 - indépendants l'un de l'autre (ils peuvent donc s'exécuter simultanément) ;
 - et n'ont pas besoin que l'envoi soit complètement initialisé pour s'exécuter ;

...ils sont directement déclenchés par le prédicat correspondant à la création de la relation Envoi|Utilise|Catégorie.

Question : Pourquoi la propriété Envoi.PoidsPli est-elle "constante" (présence du symbole (C)) ?

Réponse : parce que les calculs de nombre de plis utilisent cette donnée. Ainsi, en cas d'erreur, la seule possibilité de correction consiste à supprimer et recréer l'envoi, déclenchant ainsi à nouveau les calculs avec la nouvelle valeur.

Modèle détaillé du processus de calcul des nombres de plis par liasse



C'est un simple "éclatement" de la règle de gestion correspondante du schéma précédent.

Commentaires et implications

On voit donc que le Réseau Sémantique Universel permet de modéliser des traitements avec plusieurs niveaux de détail. On peut donc partir du principe qu'un processus (terme finalement préférable à celui d'"action" proposé dans le document de travail initial, sauf dans le cas d'un traitement élémentaire non décomposable) peut être plus ou moins détaillé, et que dans le cas d'une modélisation de haut niveau, les données strictement internes à un processus non détaillé n'ont pas à être représentées : elles apparaîtront dans le modèle décrivant le processus.

Inversement, on peut voir dans le dernier modèle que certains calculs sont soumis à une condition préalable (existence ou non d'un nombre minimal de plis par liasse au niveau Catégorie, par exemple). On peut donc se demander si on ne pourrait pas pousser un peu plus le détail en faisant de cette condition un prédicat et en décomposant le processus en deux actions élémentaires. La réponse est non, car une seule propriété est mise à jour, et il a été spécifié dans le document de travail la règle comme quoi une information ne peut être altérée que par un seul et unique processus, garantissant ainsi l'intégrité des données.

On pourra aussi constater l'apparition d'une règle importante : toute modélisation d'un processus, quel que soit son niveau, doit présenter au moins un capteur et un effecteur, correspondant aux notions d'événement et de résultat d'un MCT. En effet, toute unité de traitement est déclenchée par quelque chose provenant de l'extérieur (dans l'exemple montré, la création de la relation Envoi|Utilise|Catégorie) et fournit un résultat susceptible de déclencher un autre traitement.

Évidemment, plusieurs incertitudes sont mises en évidence (c'est le but de l'étude de cas). En dehors de l'identification relative déjà relevée, on peut se demander :

Comment prendre en compte le fait qu'une propriété non facultative peut n'être pas alimentée immédiatement par l'action de création de l'objet, mais dans un autre sous-processus du même ensemble (ceci n'est pas une remise en cause des notions de base de Merise, mais un simple effet secondaire de la décomposition des tâches qui permet une modélisation à un niveau beaucoup plus fin que le classique MCT).

Certains processus créent des objets et/ou des relations, d'autres se contentent de les compléter ou de les mettre à jour, d'autres encore effectuent des suppressions. Faut-il distinguer graphiquement ces notions pour mieux mettre en évidence les différents rôles (par un symbole qui s'ajouterait aux doubles flèches, par exemple) ?

Autre problème (peut-être simplement personnel, mais si j'ai moi-même des problèmes avec cette notion, je ne dois pas être le seul) :

Le MCD montre clairement qu'un objet Routage est identifié par l'envoi auquel il appartient. Mais il existe un autre chemin pour aller de Routage à Envoi :

Routage|Assigné_à|(Quartier|est_concerné_par|Envoi).

Ce chemin doit impérativement mener au même objet Envoi. Il faut donc préciser ceci avec une contrainte d'unicité. Quelle est la bonne représentation ?▲

Jean-Luc Blary
Consultant chez EPHITEQ
e-mail : jean-luc@blary.com



Le bogue M06215

Les risques du système métrique

Un beau jour de fin septembre 1999, la sonde Mars Climate Orbiter a prématurément interrompu sa mission, en se fracassant sur la planète qu'elle était censée observer.

Les médias, si prolixes sur les dangers du fameux « bogue » de l'an 2000, baptisé Y2K, se sont montrés beaucoup plus discrets sur les origines de cette catastrophe, qui certes n'a entraîné la perte d'aucune vie humaine, mais a volatilisé une somme « astronomique » de dollars.

Il semblerait qu'une confusion d'unités, lors de la transmission d'une donnée représentant une distance, soit à l'origine de la défaillance. L'organe destinataire du message aurait considéré que le nombre transmis mesurait la distance en miles, alors que l'émetteur l'avait exprimée en kilomètres ; à moins que ce ne soit l'inverse ! D'où une grossière erreur d'appréciation et cette manœuvre qui s'est révélée suicidaire.

Dans le cadre de la mondialisation des échanges, la juxtaposition de plusieurs systèmes de mesures nous fait courir des risques colossaux, auprès desquels ceux de l'an 2000 n'auront été qu'une émotion passagère. Ce qui s'est produit, à des milliers de miles de notre terre, peut tout aussi bien provoquer des dégâts considérables sur notre planète (divergence de réacteur nucléaire, déclenchement intempestif de riposte militaire, divagation aérienne, valse des cours de bourse, etc.). À ce propos, d'ores et déjà, les transactions sur les valeurs boursières s'effectuent en euros ; certains petits épargnants, mal informés de la cohabitation, provisoire, de deux unités monétaires, ont eu la mauvaise surprise de voir les montants de leurs ordres multipliés par le coefficient 6,55957.

De toutes les unités de mesure, créées par l'homme, ne subsistent que deux ensembles cohérents.

Le système métrique est né en France, il y a 200 ans, de la volonté de quelques théoriciens qui, sous prétexte d'améliorer le fonctionnement de la société, lui ont imposé des normes artificielles. On peut rapprocher leur utopie de celle qui préconiserait d'apprendre à parler espéranto en 1999. D'ailleurs, nombre de leurs brillantes innovations, comme le calendrier républicain, sont retombées dans l'oubli ; mais le système métrique, en absence de concurrence sérieuse sur le plan national, a échappé au couperet et survécu jusqu'à nos jours.

L'autre système (poétiquement peuplé de lignes de pouces, de pieds, de miles, d'onces, de livres, de pintes, de gallons, de pierres, de tonnes courtes et longues, de degrés Fahrenheit ...) trouve sa justification, non seulement dans la préservation d'une culture millénaire, mais surtout dans les succès scientifiques, techniques et économiques obtenus par ceux qui le maîtrisent.

Pour éviter la multiplication de telles catastrophes, il semble grand temps de procéder à un sérieux ménage dans les systèmes d'unités. Soyons pragmatiques ; de même que l'on utilise l'anglais pour les communications interrégionales, adoptons rapidement ce système anglo-saxon qui a fait largement les preuves de son efficacité, dans tous les domaines.

Et puis, après la mise à niveau des dates pour le passage à l'an 2000, la mise aux normes de toutes les mesures, dans les programmes informatiques, procurera un travail substantiel à nos informaticiens.

*Alain Coulon
Secrétaire d'ADELI*

☛ M06215 : Un kilomètre équivaut à 0,6215 euromile

☛ Attention à une autre confusion - cet article est entièrement écrit au second degré.