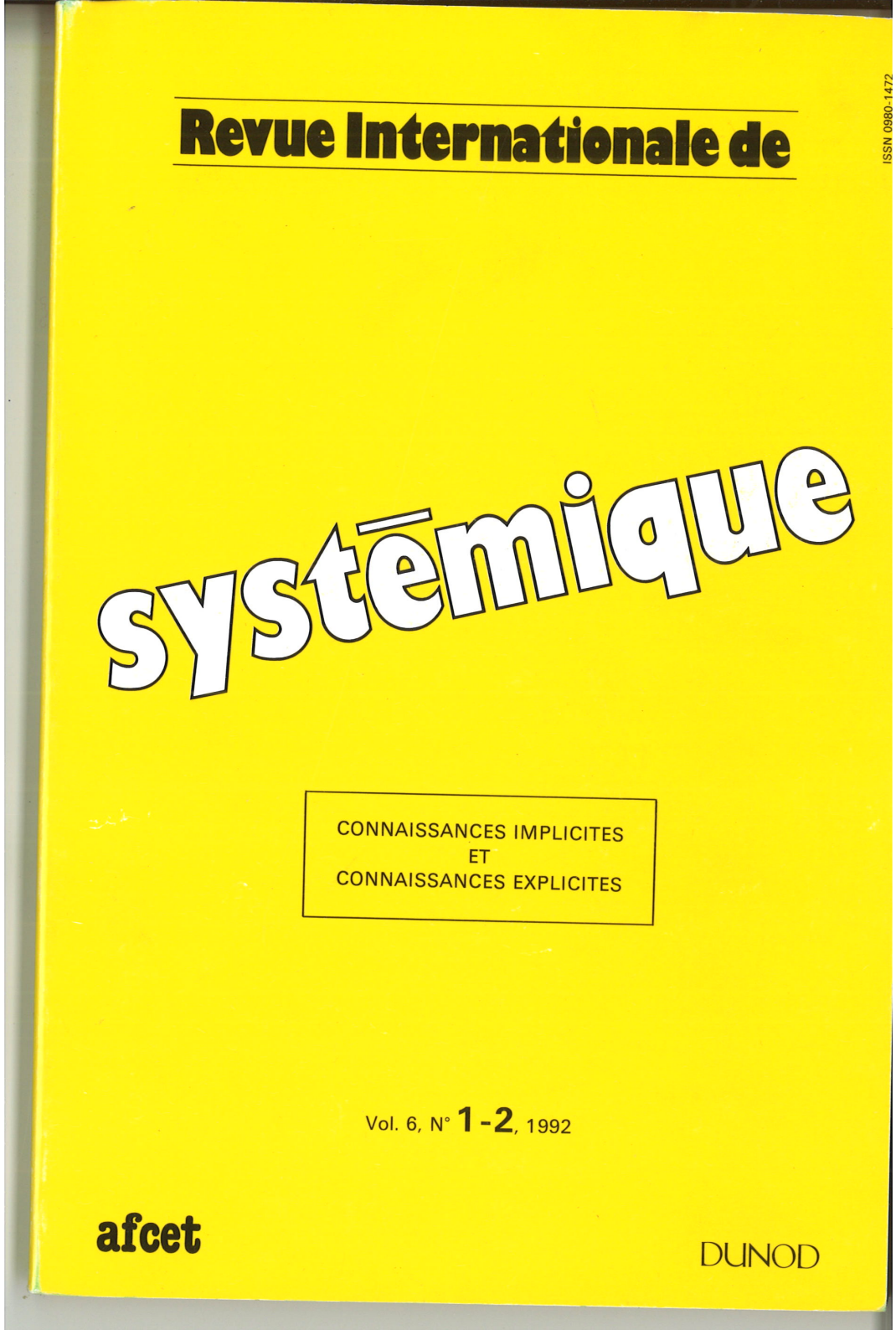


AFSCET

Revue Internationale de systēmique

Revue
Internationale
de Sytémique

volume 06, numéro 1-2, pages 47-61, 1992

Enseignements méthodologiques issus de la réalisation d'un système à base de modèles pour la gestion du trafic routier

Gérard Scemama

Numérisation Afscet, août 2017.

Creative Commons

REVUE INTERNATIONALE DE SYSTÉMIQUE
Vol. 6, N° 1-2, 1992, pp. 47 à 61

ENSEIGNEMENTS MÉTHODOLOGIQUES ISSUS DE LA RÉALISATION D'UN SYSTÈME À BASE DE MODÈLES POUR LA GESTION DU TRAFIC ROUTIER

G. SCEMAMA

Institut National de Recherche
sur les Transports et leur Sécurité [1]

Résumé
Cet article présente les principaux enseignements méthodologiques issus de la réalisation d'un système expert pour la gestion du trafic routier. L'approche cognitive fondée sur le transfert de connaissances d'experts humains est comparée à une démarche d'élaboration de Systèmes à Base de modèles (SBM) s'appuyant sur des connaissances explicites du domaine. On montre notamment que la première approche pose un grand nombre de problèmes de génie logiciel.

Abstract
This paper presents main methodological characteristics issued from the development of an expert system for road traffic management. The cognitive approach based on knowledge acquisition from human experts is compared with a modelling approach using explicit knowledges in the field. It is shown that the first approach raises a number of problems related to software engineering.

Cet article présente un certain nombre d'enseignements méthodologiques sur le développement de Systèmes Experts (SE). Le point de vue mis en avant consiste à opposer l'approche cognitive de transfert de connaissances d'experts humains à une démarche de construction de modèles qualitatifs s'appuyant sur des connaissances explicites du domaine. On parlera alors de conception de Systèmes à Base de Modèles (SBM).

1. Dept. MAIA, 2, avenue Général Malleret Joinville, B.P. 34, 94114, Arcueil Cedex, France.

Rev. intern. systémique. 0980-1472 Vol. 6/92/01-2/ 47 /15/$ 3.50/© Afcet Gauthier-Villars

La distinction est d'importance lorsqu'on est soumis à la contrainte de développer une *application informatique*, car elle induit des différences d'ordre méthodologique non négligeable.

Dans le premier cas, on cherche à capturer les connaissances heuristiques des experts. On se heurte aux problèmes du transfert de ces connaissances (connaissances implicites). Cette approche empirique cadre mal avec les contraintes liées à tout développement logiciel. Elle ne permet pas de combler le fossé conceptuel séparant la connaissance d'un expert humain de la réalisation d'un programme informatique.

Dans le second cas, on cherche à élaborer un modèle non-numérique du monde réel. On parle alors plus volontiers de Systèmes à Base de Modèles (SBM). Une telle approche permet d'entrevoir une méthodologie de l'ingénierie de la connaissance qui s'inscrit dans le cadre de l'informatique conventionnelle. Le cycle de développement d'un SBM peut être articulé en phases qui séparent notamment l'analyse, de l'implémentation.

Les principales réflexions se fondent sur la réalisation du système SAGE (Système d'Aide à la Gestion des Embouteillages) (Forasté et Scemama, 1986), développé pour résoudre les problèmes de circulation automobile. Ce système est passé successivement du stade de la maquette de laboratoire en 1985, à celui de prototype expérimental sur site en 1987, pour atteindre finalement un stade opérationnel en 1990 à Paris. Les recherches autour de ce système se poursuivent dans le cadre du programme DRIVE (Dedicated Road Infrastructure for Vehicule safety in Europe) lancé par la Communauté Économique Européenne (Schnetzler *et al.*, 1990), (Bell *et al.*, 1991). Compte tenu de l'importance qu'a revêtu ce développement, il apparaît important de préciser les quelques points clés qui ont concouru à l'élaboration de ce système et par voie de conséquences à la maîtrise d'une technique.

1. Le système SAGE

Les systèmes de régulation de trafic automobile ont pour fonction de traiter les données en provenance des capteurs et d'agir en retour sur la commande des feux de signalisation. Les automatismes (sélection et/ou adaptation temps réel des plans de feux) ne suffisent pas en cas de congestion. Ils reposent sur des hypothèses de fluidité du trafic et travaillent sur des données supposées connues de façon certaine. Or un système paraît d'autant plus complexe que l'information est incertaine et incomplète. Les centres de régulation ont donc recours à des équipes spécialisées (opérateurs, ingénieurs de la circulation) qui ont la charge de surveiller, d'analyser, d'étudier et de prendre les mesures les plus appropriées.

Le système SAGE est un composant d'un système de régulation. Il assure en temps réel les fonctions de surveillance, de diagnostic des causes de congestion et de commande de plans de feux. Il repose sur un modèle causal symbolique des phénomènes de congestion et de leurs procédures de traitement. Le recours à l'Intelligence Artificielle a été identifié pour la formalisation de ce modèle. Les aspects techniques qu'il a fallu résoudre concernent ceux liés au développement d'un Système à Base de Connaissances temps réel ayant à gérer un historique important (raisonnement temporel, information incomplète, problème combinatoire, problème de performance, de stockage de l'information...). L'aspect combinatoire de l'application nous a conduit au début au choix du moteur SNARK (Laurière, 1986*a*, *b*) qu'il a fallu adapter au fonctionnement continu. Enfin, ces travaux ont débouché sur la réalisation d'un nouveau moteur d'inférences AIDA (Schnetzler, 1988). Ce langage permet la formalisation d'énoncés logiques compliqués et gère l'aspect temporel.

La réalisation de SAGE a fait apparaître la nécessité de mettre au point une méthodologie de développement de SBM compatible avec un développement industriel et dont nous décrivons les principales lignes dans la suite de ce papier.

2. Méthodologies de développement

Face au développement de ce système, deux options méthodologiques étaient envisageables : l'une fondée sur le transfert des connaissances de l'expert humain, l'autre reposant sur l'élaboration de modèles.

2.1. *Méthodologie fondée sur le transfert des connaissances*

Cette première méthodologie s'appuie sur l'analyse de l'activité humaine et dans la reproduction des raisonnements observés sur un ordinateur. Un certain nombre de travaux (Bonnet *et al.*, 1986) ont tenté de dégager une méthodologie générale de développement de système fondée sur ce principe.

Nous allons montrer les conséquences d'une telle approche à partir du texte intitulé *Traitement de la maille Victoria* décrivant des heuristiques mises en œuvre par les ingénieurs de trafic pour faire face à des phénomènes de congestion. Ce texte fait partie des premiers documents rédigés par les Ingénieurs de la Ville de Paris. Les procédures décrites cherchent à protéger cette maille critique contre les remontées de queues provenant des axes Rivoli et/ou Sébastopol. Sa lecture montre les diagnostics de congestion établis à partir des observations de trafic (postes de mesure de Taux d'Occupation « TO », Débit) ainsi que certaines actions, sous la forme de plans de feux

« PF » lancées pour y remédier. L'énoncé de ces pratiques se prête aisément à leurs représentations sous forme de règles de production. La connaissance exprimée est relative à des chaînons et détecteurs bien spécifiés. Cependant, la compréhension des méthodes de traitement de la maille Victoria dans ce texte repose sur la connaissance du domaine de la régulation du trafic et des mécanismes de propagations de la saturation dans un réseau (connaissances implicites).

Le développement d'un système expert fondé sur ces heuristiques est tout à fait envisageable. Il conduit à l'élaboration d'un système certes efficace, mais faisant état de connaissances superficielles, reflétant les connaissances de l'exploitation d'un réseau particulier. Cette approche possède en outre de sérieuses limitations. Elle serait dépendante des caractéristiques de la circulation de la voirie Parisienne. Toute nouvelle évolution des conditions de circulation remettrait en cause en partie le système développé.

Clancey (1983) fait un constat identique à propos du système de diagnostic médical MYCIN. Il met en évidence la faiblesse de la représentation qui n'inclut pas de connaissances structurelles et stratégiques, seules aptes à fournir une véritable explication des raisonnements. Il conclut en la nécessité de rendre explicite le modèle causal sous-jacent et pose ainsi les bases d'une ingénierie de la connaissance.

L'élaboration d'un système reproduisant l'activité de l'opérateur se heurte aussi au problème de la dépendance de cette activité vis-à-vis de l'environnement des équipements disponibles. Avec de nouveaux outils (interface opérateur, caméras, moyens de mesures, ...) ces raisonnements pourront évoluer. Il n'est pas judicieux de chercher à reproduire un comportement qui peut être limité faute de moyens.

Cette démarche de transfert des connaissances effectue une rupture par rapport aux méthodes conventionnelles de modélisation et d'optimisation. Le recours à l'empirisme est interprété comme un certain recul et se trouve vivement critiqué. Les nombreux échecs constatés proviennent de la difficulté à reproduire des raisonnements humains sur un ordinateur et à une sous-estimation de la complexité des problèmes abordés.

Les heuristiques des experts humains sont des raccourcis de lois dont il nous faut trouver les fondements. Une méthodologie fondée *uniquement* sur leur codification est à exclure pour le développement d'applications informatiques. Elle peut par contre se justifier si on poursuit en soit un but d'acquisition de connaissances : l'objectif étant alors d'analyser pour mieux comprendre l'activité humaine (analyse de la tâche). On met en évidence certains mécanismes de résolution sans se soucier de leur efficacité. On peut

TRAITEMENT DE LA MAILLE VICTORIA

Remarques liminaires

Les procédures décrites plus loin essaient d'utiliser l'arsenal existant de plans de feux susceptibles d'agir sur la maille. Actuellement cet arsenal n'est pas utilisé systématiquement. Cette utilisation est fonction de l'opérateur qui peut, soit en faire trop, soit pas assez. De toute façon, il agit la plupart du temps trop tard.

L'action est souvent déclenchée lorsque le synoptique montre un engorgement de Renard ou Coutellerie (clignotant du voyant de charge sur le synoptique) ce qui, compte tenu du lissage sur 12 minutes et de l'inertie du traitement de l'affichage impose fréquemment, une mesure énergique (instruction 102 ou forçage du PF (Plan de Feux) 10 sur la zone 1).

La mise au clair du traitement de cette maille fait apparaître pour améliorer le traitement de la saturation les besoins suivants :
— affinage des diagnostics. Les seuils de décision cités seront à réétudier plus finement;
— création des plans de feux spéciaux pour cette maille qui, en fonction de quelques traitements types et *gradués* simplifieraient le processus. Il faut signaler que, parallèlement, est envisagée une nouvelle politique de traitement de la saturation ponctuelle qui nécessite une refonte du logiciel actuel;
— d'une façon plus générale, redéfinition des objectifs, et des mesures à prendre en matière de saturation.

001 Si un TO = Taux d'Occupation TO ⩾ 30% est détecté sur au moins un des postes 13 et 14 de l'itinéraire (Sébastopol) et/ou un TO ⩾ 75% est détecté sur 2 postes 10 à 14 de l'itinéraire 3 (Rivoli) attirer l'attention de l'opérateur et demander l'observation de l'intersection Sébastopol Victoria par la caméra 09.

002 Y a-t-il un spécial sur la zone? Si oui appliquer III.

I. Le PF 9 (journée) est en cours sur la zone 1

003 Si le diagnostic 001 est vérifié et si Rivoli entre Sébastopol et Renard sature (TO des postes 8 et 9 ⩾ 15%) dégager Renard (instruction 100) — Sauvegarde Rivoli/Renard

004 Si le même phénomène existe sans le diagnostic 001, demander l'accord de l'opérateur pour appliquer l'instruction 100 — Cette instruction est discutable

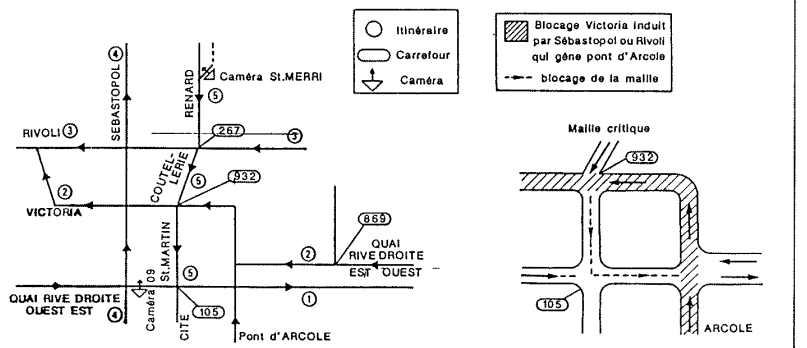
005 Si le diagnostic 001 est vérifié et si l'amont de Victoria est gêné par Sébastopol (TO poste 8 de l'itinéraire 2 ⩾ 20%, dégager Coutellerie (instruction 101) — Sauvegarde Coutellerie, Renard 1re étape

006 Si l'amont de Victoria est gêné par Sébastopol mais que le diagnostic 001 n'est pas établi, demander l'accord de l'opérateur pour dégager Coutellerie (instruction 101) — Instruction discutable

007 Si le diagnostic 001 est établi et que :
— Victoria est bloqué par Sébastopol (TO poste 8 de l'itinéraire 2 ⩾ 30%;
— Coutellerie a un TO ⩾ 30% (poste 2 itinéraire 5)
— le débit de Renard est < 1 200 dégager énergiquement Coutellerie (instruction 102) — Sauvegarde Coutellerie, Victoria 2e étape 9/1

008 Si l'une au moins des conditions du diagnostic 007 n'est pas vérifiée, et que le diagnostique 005 ou 006 est établi proposer d'appliquer l'instruction 102 en demandant de vérifier :
— la caméra 09 (blocage)? — possibilité d'agir en temps réel
— Victoria amont Sébastopol (itinéraire 2) remplie? — anticipation possible de 007
— TO Coutellerie: Coutellerie est-elle remplie?
— débit Renard; le débit chute-t-il?

009 Si sur les quais Est-Ouest (itinéraire 2) le TO 7 (place H.V) ⩾ 20% retenir à Lobau (instruction 103) — sauvegarde de la maille quoiqu'il arrive



développer les programmes correspondants. Mais ces programmes n'ont pas vocation à être des programmes de résolution du problème mais de simulation de certains comportements face à ce problème.

2.2. *Méthodologie fondée sur l'élaboration de modèles*

Cette méthodologie consiste à élaborer un *model causal* qualitatif du phénomène que l'on veut étudier. Un tel modèle s'appuie sur des connaissances explicites du domaine et propose un représentation causale du monde réel. Le modèle contient une interprétation sur le comportement et l'évolution du phénomène et constitue une théorie scientifique sous-jacente. Un exemple de tels modèles est présent en physique qualitative (*cf*. le numéro spécial d'*Artificial Intelligence*, Bobrow, 1984; Forbus, 1984; Kleer et Brown, 1984; Kuipers, 1984) mais nous l'entendons dans un sens plus général. Cette approche ne se démarque pas des méthodes quantitatives de modélisation si ce n'est par la nature des traitements qui seront symboliques. Son avantage est qu'elle s'inscrit dans une démarche positive d'apport de connaissances au domaine. Ses difficultés résident dans la nécessité d'avoir à élaborer le dit modèle.

Un de ses autres intérêts est qu'elle permet de fixer le choix des applications à développer. Il faut aborder des problèmes de complexité raisonnable. S'agissant de problèmes devant être résolus par la machine (Laurière, 1986*a*, *b*) il faut que la solution soit fournie en un temps acceptable. A cet égard, il n'est pas interdit d'utiliser des heuristiques pour accélérer l'obtention d'une solution du problème. Mais ces heuristiques là interviennent dans une méthode de résolution déjà formalisée et donc différent fondamentalement des précédentes. Les spécialistes du domaine (experts) restent une référence essentielle. Ces derniers interviennent dans le processus d'élaboration du modèle et ne sont pas seulement les dispensateurs d'un « savoir-faire ».

Cette approche permet d'envisager la réalisation d'un système expert dans un cadre plus conventionnel. Nous exposerons dans la suite les différentes phases que nous avons identifiées comme devant intervenir dans le cycle de développement. La séparation entre l'analyse et la programmation permet de valider les modèles logiques indépendamment des outils choisis pour l'implémentation. Enfin, l'élaboration de modèles explicites peut constituer la base d'une documentation qui permet d'envisager plus facilement *la maintenance* et *l'évolution* du système.

2.3. *Comparaison des approches*

Quelques différences entre ces deux approches sont récapitulées dans le tableau ci-dessous. En réalité, deux options sont apparues dès le début du développement du système : soit de partir directement des descriptions des ingénieurs de trafic, soit d'essayer de formaliser les concepts généraux.

Approche fondée sur le transfert de connaissances	Approche fondée sur l'élaboration d'un modèle
Définition	
Analyse de l'activité humaine, reproduction de la démarche d'un expert humain.	Élaboration d'un modèle qualitatif causal du monde réel.
Concept de transfert et d'acquisition des connaissances.	Représentation et structuration des connaissances (modèle de données).
	Élaboration des modèles logiques.
Problèmes	
Définition de l'expert. Comment extraire la connaissance ?	Trouver le modèle efficace (complexité raisonnable).
Sous-estimation de la complexité du problème (connaissances implicites).	Difficulté à généraliser et à abstraire les concepts.
Rejet des connaissances empiriques (connaissances superficielles, concepts parfois non partagés).	Apport des connaissances dans le domaine.
Validation	
Comparaison des conclusions du système et de l'expert humain.	Consensus sur le modèle (reproductibilité des raisonnements).
Profil du développeur	
Connaissance en psychologie et en sciences de la communication.	Aptitude en logique, en modélisation, Connaissances dans le domaine.

Le développement étant entrepris dans un contexte de recherche, la seconde voie s'imposait. La réalisation du système spécifique à une ville donnée ne rentrait pas dans les missions d'un Institut de recherche (INRETS). Cependant pour des raisons de coût du développement), une telle option n'est souvent pas prise par les développeurs d'applications. La première option (transfert de connaissances) s'appuie sur une méthodologie dite du *prototypage rapide*. On aboutit à un système de portée limitée qui ne traite qu'une partie du problème, qui n'est en rien la démonstration d'une faisabilité et qui est très vite déprécié.

3. Ingénierie du système à base de modèles

Une méthodologie de développement associée à la réalisation de ce type de système a été définie. Le cycle de développement considère les phases suivantes :

- l'identification des fonctions du système à réaliser,
- l'élaboration des modèles de données,

- la formalisation des modèles logiques,
- la validation,
- la programmation des règles dans le langage approprié,
- le test et l'évaluation.

3.1. *Identification des grandes fonctions du système*

Cette première étape dans le cycle de développement du logiciel concerne l'identification des grandes fonctions que l'on veut réaliser.

Les grandes fonctions envisagées dans SAGE sont la surveillance, le diagnostic, la mémorisation des événements, le traitement de la congestion.

Cette étape n'est pas forcément évidente (surtout dans un contexte de recherche), elle doit permettre de conclure en partie sur le bien-fondé du choix de l'application.

3.2. *Élaboration des modèles de données*

L'élaboration d'un système à Base de Modèles ne diffère pas tellement de celle d'un Système de Gestion de Base de Données (SGBD) (Gardarin et Valduriez, 1989), en ce sens que l'on est amené à faire la description des objets du monde réel. Cette phase consiste en la description de toutes les entités et associations. La transition du monde réel au niveau conceptuel correspond à un premier pas vers la modélisation et la structuration des connaissances. On est conduit à définir le schéma conceptuel. Un modèle de données possible pour décrire un tel schéma est le modèle relationnel ou le modèle entité-association (Chen, 1976).

Les objets de notre univers de description (la base de connaissance) sont représentés par des relations sur des ensembles :

$$R = \{ (x_1, x_2, \ldots, x_n) / x_1 \in X_1, x_2 \in X_2, \ldots, x_n \in X_n \}$$

Les ensembles X_1, X_2, X_n sont appelés domaines de la relation tandis que les composantes de la relation sont appelés attributs.

Par convention, nous pourrons écrire :

$$R(\text{attribute}_1, \text{attribute}_2, \ldots, \text{attribute}_n)$$

$$(\forall r \in R)\, r(x_1, x_2, \ldots, x_n) \Leftrightarrow (\forall r \in R)(\forall i \in \{1, \ldots, n\})\, r.\text{attribute}_i = x_i.$$

Cette équivalence traduit la décomposition d'une relation n-aire en n relations binaires.

Notons qu'un vecteur $r(x_1, x_2, \ldots, x_n)$ n'est pas nécessairement une structure « plate ». Un attribut peut ne pas être atomique et se référer à d'autres

relations. Ainsi quand X_i est le domaine d'une relation :

$$x_i \in X_i \Rightarrow x_i = (y_1, y_2, \ldots, y_m) \Rightarrow r.\text{attribute}_i = (y_1, y_2, \ldots, y_m)$$

Afin de reconnaître les vecteurs de différentes relations, nous utilisons la notation suivante :

$$(\forall r \in R)\, r(x_1, x_2, \ldots, x_n) = R(x_1, x_2, \ldots, x_n)$$

```
MODELE DE DONNEES[1]

NŒUD  (nom, x, y,  classe)

        type CLASSE-NŒUD is enumerated of
                (intersection, entree, sortie).

        type NŒUD is relation (nom) ;
                nom             : identifier
                x               : float.    ; abscisse du carrefour
                y               : float.    ; ordonnée du carrefour
                classe          : CLASSE-NŒUD.
        end relation.

CHAINON  (nom, origine,  extremite, capacité, état).

        type ETAT-CHAINON is enumerated of
                (fluide, sature, inconnu).

        type CHAINON is relation (nom)
                nom             : identifier.
                origine         : NOEUD-KEY. ; accés à la clé du noeud
                extremite       : NOEUD-KEY.
                capacité        : VEHICULE/HEURE.
                etat            : ETAT-CHAINON.
        end relation.

AXE  (nom,  chainons)

type LIST-CHAINON-KEY is list of CHAINON-KEY.

        type AXE is relation (nom)
        nom             : identifier.
        chainons        : LIST-CHAINON-KEYS.
        end relation.

DONNEES ou PREDICATS

        NŒUD (c1 , 1.015, 8.098,  carrefour).
        CHAINON  (C256, C1, rivoli-1, 2500.0, fluide).
        AXE (Rivoli, (rivoli-10, rivoli-892, rivoli-167, rivoli-1, rivoli-256, rivoli-267 ) ).
```

1. En souligné les attributs faisant partie de la clé.

Compte tenu du caractère dynamique de l'univers modélisé, nous avons introduit un étiquettage temporel des valeurs des attributs d'une relation par leur date d'apparition :

$$(\exists r \in \mathrm{R})(\exists t1, t2, \ldots, tn \in \mathbb{T})$$
$$r(x1_{t1}, x2_{t2}, \ldots xn_{tn}) = \mathrm{R}(x1_{t1}, x2_{t2}, \ldots, xn_{tn})$$

où *ti* représente la date à laquelle le *n*-ième attribut de R a pris la valeur *xi*.

Quand les dates des valeurs prises par les attributs d'une relation sont toutes identiques on peut factoriser cette date :

$$(\exists r \in \mathrm{R})(\exists t \in \mathbb{T})\, r_t(x1_t, x2_t, \ldots, xn_t) = \mathrm{R}_t(x1, x2, \ldots, xn)$$

La nécessité d'avoir à décrire des structures de données complexes nous a amené à enrichir le modèle relationnel avec l'élaboration d'un Langage de Description des Données (LDD), comme extension du modèle relationnel. Il permet de définir le schéma conceptuel de la base de connaissances en terme de relations CHAINON, CARREFOUR, AXE, ...

Les applications développées font en général appel à la fois à des composants algorithmiques et déclaratifs. Il faut éviter de déclarer deux modèles réalisant la même description. Le LDD permet des déclarations de type similaires à celle des langages classiques (Pascal, Ada, ...).

Cette étape de structuration explicite de la connaissance est particulièrement importante dans la conception du système. Nous présentons ci-dessous quelques-unes des déclarations de modèles ainsi que les données ou tuples associés.

3.3. *Formalisation des modèles logiques*

En terme de méthodologie de développement, nous séparons la description des modèles logiques de leur programmation dans le langage choisi.

Cette étape consiste en la description (en utilisant la logique) du processus modélisé (règles de causalité) et des fonctions de traitement (règles de diagnostic). Elle nécessite la maîtrise de la logique. Cette phase est analogue à celle de définition des algorithmes dans un projet faisant appel à une modélisation numérique.

Les énoncés logiques du modèle doivent être décrits de façon précise se rapprochant d'une description formelle. Cela nous a amené à définir une notation logique compatible avec l'utilisation des modèles de données.

On peut en effet interpréter les instances d'une relation comme une expression de la logique. Ici on fait référence à une logique temporelle qui est une extension de la logique du premier ordre.

Ainsi l'expression ci-dessous :

$$(\exists r \in \mathrm{R})(\exists t1, t2, \ldots, tn \in \mathbb{T})$$
$$r(x1_{t1}, x2_{t2}, \ldots, xn_{tn}) = \mathrm{R}(x1_{t1}, x2_{t2}, \ldots, xn_{tn})$$

correspond à un énoncé logique exprimant la véracité du fait « chaque attribut prend la valeur *xi* entre *ti* et l'instant courant ».

Pour la fonction de diagnostic de SAGE, un modèle qualitatif d'interprétation de la congestion a été réalisé. Il repose sur l'élaboration de « chaînes saturées » (chemin de chaînons saturés d'un même axe de circulation) à partir de données qualitatives sur les états de chaînon (fluide, saturé, inconnu).

Le modèle CONGESTION décrit une congestion par son carrefour d'origine, l'axe sur lequel elle s'est propagée et les chaînons qui la composent.

Le modèle CAUSE permet de relier deux saturations sur deux axes différents dans l'association victime, responsable.

```
MODELE DE DONNEES

CONGESTION (origine, axe, chainons)

        type CONGESTION is relation (origine, axis)
                origine         : NOEUD-KEY.
                axe             : AXE-KEY.
                chainons        : LIST-CHAINON-KEY.
    end relation.

CAUSE (victime, responsable)

        type CAUSE is relationship between
                victime         : a CONGESTION.
                responsable     : a CONGESTION.
        end relationship.

DONNEES

        ; saturation sur Rivoli au départ de C1 ayant pris naissance à l'instant t1 et qui s'est rallongée
successivement aux instants t2, t3.
        CONGESTION (C10t1, Rivolit1, (rivoli-10t1, rivoli-892t2, rivoli-167t3, rivoli-1t3) ).

        ; saturation sur Victoria  au départ du noeud C167 ayant pris naissance à l'instant t2
        CONGESTIONt2 (C167, Victoria, (Victoria-167,  Victoria-632, Victoria-932)).

        ; cause indiquant que le responsable de la saturation de Victoria résulte de la saturation sur Rivoli
depuis l'instant t2
        CAUSEt2 ((C167, Victoria), (C10, Rivoli)).
```

Ces saturations sont ensuite reliées entre elles par un schéma de causalité. Le modèle utilise la propriété physique que la saturation se propage de l'aval vers l'amont en respectant une contrainte de précédence temporelle.

ÉNONCE LOGIQUE

Soient 11, 12 deux chaînons successifs d'un même axe si 11 et 12 sont saturés à des instants satisfaisant une contrainte de précédence temporelle alors ils appartiennent à une même congestion.

Soit formellement :

$$(\forall 11, 12 \in \text{CHAINON})(\exists \delta \geqq 0)(\exists t1, t2 \in \mathbb{T})(t1 \leqq t2 + \delta)$$
$$(\exists a \in \text{AXE})\, a(_, \ldots, 11, 12, \ldots) \wedge (11.\text{etat}_{t1} = 12.\text{etat}_{t2} = \text{saturé})$$
$$\Rightarrow$$
$$(\exists c \in \text{CONGESTION}\, c(_, _(\ldots, 11_{t1}, 12_{t2}, \ldots)))$$

Toute cause de saturation prenant naissance à un carrefour donné est transmise aux chaînons amont suivant ce principe. Le système est alors apte à suivre l'évolution de la saturation dans le temps et dans l'espace. Cependant ce modèle reste une approximation de la réalité et certaines déductions peuvent ne pas correspondre au phénomène physique qui se déroule.

3.4. *Validation*

La validation est l'adéquation du modèle par rapport au monde réel (retour au concret). Les travaux en théorie de la démonstration (Herbrand) ont contribué à faire apparaître deux notions fondamentales : la décidabilité et l'interprétation. La décidabilité est la possibilité de démontrer un théorème comme une suite finie d'axiomes, tandis que l'interprétation est une mise en rapport d'une théorie avec l'univers. Ainsi un énoncé peut être ou ne pas être un théorème (décidabilité), peut être ou ne pas être vrai (interprétation).

Dans le cas des systèmes s'appuyant sur la logique d'ordre 1 (non décidabilité), la validation ne peut reposer que sur un *consensus* sur la théorie sous-jacente développée. Comme en Sciences Expérimentales, cela signifie :

(*i*) l'acceptation du point de vue formel des axiomes, hypothèses et règles de dérivation (reproductibilité des raisonnements) mais aussi;

(*ii*) la vérification par l'expérience des hypothèses et conclusions produites. Le jour où on met en évidence une contradiction, on devra revoir la théorie en question.

La confrontation avec les spécialistes du domaine peut être envisagée à ce niveau.

Dans peu de cas, la validation est envisagée sous cet angle. Les raisons invoquées reposent sur le fait que les systèmes experts ont des spécifications incomplètes, manipulent des connaissances empiriques et sont non déterministes. Ils échapperaient donc au test d'adéquation, la validation se limitant en des tests de concordance avec les conclusions des experts humains (*ii*).

De notre point de vue cela n'est pas acceptable car le consensus doit reposer sur la compréhension des raisonnements implémentés (*i*). Cette dernière constatation ne concerne pas les systèmes fondés sur des modèles explicites et doit nous interpeller quant à la validité des systèmes fondés sur la capture d'heuristiques sans inclure de connaissances structurelles et explicatives. On conçoit donc l'avantage de notre approche où comme en informatique classique, les fonctions du système sont spécifiées, la validation des modèles logiques entreprises avant toute programmation.

3.5. *Programmation des règles*

La programmation consiste en l'écriture des règles de production correspondant au modèle logique formalisé dans l'étape précédente. Il n'y a pas forcément une correspondance biunivoque entre les énoncés logiques et leurs traductions en règles de production. Il faut se plier aux possibilités d'expression logique du moteur d'inférences. Par exemple, plusieurs règles de production sont parfois nécessaires pour traduire un énoncé logique.

Il faut aussi définir les principales phases d'enchaînement des processus de traitement. L'expérience montre que l'enchaînement des diverses fonctions du système est défini d'une manière relativement impérative.

Les phases de spécification ont permis de définir un système logiquement sans se préoccuper des contraintes matérielles.

Un ordinateur ne pouvant pas mémoriser indéfiniment des informations, on doit à un certain moment éliminer certaines données devenues obsolètes. Le critère retenu est de faire définir par l'application quelles informations éliminer et quand les éliminer.

La limitation de puissance de l'ordinateur pose le problème d'une évaluation efficace des règles. Cependant, l'optimisation doit de façon impérative être indépendante des spécifications logiques (écriture des règles). L'évaluation d'une règle est du ressort du moteur d'inférences, elle peut être optimisée à partir d'informations fournies par l'utilisateur sans répercussions sur le corps de la règle. Il ne faut pas que des considérations de contrôle viennent dénaturer l'expression logique d'une règle.

3.6. *Test et évaluation*

Il y a lieu de distinguer l'évaluation de la validation. Alors que la validation consiste à établir la cohérence et l'adéquation des modèles proposés, l'évaluation vise à mesurer l'efficacité du système dans son contexte opérationnel.

L'évaluation d'un SBM peut être conduite en considérant :
- l'efficacité de sa relation avec l'utilisateur final;
- l'efficacité dans le traitement du problème.

Le premier critère repose sur le fait que le SBM reste un programme informatique. A cet égard, il n'échappe pas aux critères relatifs à l'évaluation de la qualité des logiciels comme les temps de réponse, la qualité des interfaces, la documentation, etc.

Le second critère repose sur le fait que le système effectue un certain traitement dont on attend qu'il soit efficace (ici le traitement de la congestion). L'analyse statistique est la technique utilisée pour mener à bien cette évaluation.

Une mesure de l'efficacité du système a été entreprise dans le cadre du test du système SAGE (enquêtes par photographies aériennes, véhicules flottants, ...) en 1990. Les résultats ont conduit la Ville de Paris à retenir le système dans son projet d'extension de la gestion centralisée des feux.

4. Conclusion

Ce papier a montré que la méthodologie fondée uniquement sur la codification d'heuristiques d'experts humains semble peu compatible avec le développement d'un logiciel informatique. De nombreux problèmes restent posés comme celui de la formalisation des connaissances implicites. L'absence de modèles explicatifs des raisonnements (spécifications incomplètes, systèmes empiriques, non déterministes) réduit la validation au seul test de concordance avec les conclusions de l'expert humain. Il est alors difficile faire évoluer ces systèmes. L'introduction d'heuristiques n'est envisageable qu'une fois la méthode de résolution complètement formalisée. Dans ce cas, ces dernières ne sont qu'un moyen d'accélérer l'obtention d'une solution dans un cadre théorique dont la nature peut n'avoir aucun rapport avec la façon de procéder de l'expert.

L'expérience acquise dans la réalisation du système SAGE a permis d'aboutir à un concept d'ingénierie logicielle de la connaissance. Celle-ci repose sur l'élaboration d'un modèle causal symbolique du monde réel. Le cycle de développement peut être articulé en phases qui sépare l'analyse de l'implémentation. La validation se fonde sur l'adéquation du modèle par rapport au monde réel (conformité du processus modélisé, reproductibilité des raisonnements). Elle peut être envisagée avant la programmation, indépendamment du langage choisi. Cette formalisation sert de base à une documentation et permet d'entrevoir une maintenance et une évolution du système.

Remerciements

Les idées exprimées dans cet article sont le fruit d'une démarche entreprise en commun avec mon collègue Bernard Schnetzler, Ingénieur de Recherche au Centre Informatique de Recherche de l'INRETS et auteur du langage AIDA que je tiens à remercier particulièrement. Mes remerciements vont aussi à mes relecteurs dont les commentaires pertinents m'ont permis d'améliorer cet article.

Références

M. BELL, G. SCEMAMA, L. IBBETSON, Claire: an expert system for congestion management, *Drive Conférence*, Bruxelles, février 1991.

D. BOBROW, Qualitative Reasonning about physical system: An introduction, *Artificial Intelligence*, **24**, 1984.

A. BONNET, J. P. HATON, J. M. TRUONG NGOC, *Systèmes Experts : vers la maîtrise technique*, InterEditions, 1986.

CLANCEY, The Epistemology of a rule-based Expert System: a framework for explanation, *Artificial Intelligence*, **20**, (3), May 1983.

CLANCEY, Heuristic Classification, *Artif. Intell.*, **27**, (3), Dec. 1985.

P. S. CHEN, The Entity Relationship Model: Toward a unified view of Data, *ACM Transactions on Database Systems*, **1**, 1976.

B. FORASTÉ, G. SCEMAMA, Système expert temps réel d'aide à la gestion du trafic, Actes des 6 Journées Internationales, *les systèmes experts et leurs applications*, Avignon, avril, 1986.

K. D. FORBUS, Qualitative process theory, *Artificial Intelligence*, **24**, 1984.

G. GARDARIN, P. VALDURIEZ, *Databases and Knowledge bases*, Addison Wesley, 1989.

J. KLEER, J. S. BROWN, Qualitative physics based on confluences, *Artif. Intell.*, **24**, 1984.

B. KUIPERS, Commonsence reasonning about causality: deriving behaviour from structure, *Artificial Intelligence*, **24**, 1984.

J. L. LAURIÈRE, Un langage déclaratif : SNARK, *T.S.I*, **5**, (3), 1986 *a*.

J. L. LAURIÈRE, *Intelligence Artificielle, Résolution de problèmes par l'homme et la machine*, Eyrolles, Paris, 1986 *b*.

G. SCEMAMA, *Réalisation d'un Système à Base de Connaissance pour la gestion du trafic urbain*, Recherche Transport Sécurité, juin 1989.

B. SCHNETZLER, AIDA : un langage (Informatique) déclaratif, *Rapport INRETS*, n° 55, janvier, 1988.

B. SCHNETZLER, G. SCEMAMA, B. FORASTÉ, SAGE, un système à Base de Connaissances temps réel, Conférence les *Systèmes Experts et leurs Applications*, Avignon, 1990.