



Chapitre II : modèle d'indexation par situations comportementales

-
-
-
-
-
-
-
-
-
-
-
-

Résumé

Nous abordons le problème de l'indexation pour une classe de problèmes dans laquelle les indices des cas doivent reposer sur des historiques. Peu de travaux existants en RàPC ont abordé cette problématique et ils sont généralement dédiés à des applications particulières. De plus, les approches de reconnaissances d'historiques et les systèmes de gestion de bases de données temporelles ne prennent pas en compte les besoins spécifiques liés à la réutilisation et à la mise à jour des connaissances durant un cycle de raisonnement. Nos apports se situent à deux niveaux. Premièrement, nous proposons alors le « *modèle d'indexation par situations comportementales* » qui est la seule approche générique en RàPC pour l'indexation par historiques. Notre approche comprend le modèle de représentation proprement dit des indices, et un guide d'utilisation. Dans ce modèle, les enregistrements contiennent, dans des chroniques, les observations effectuées suivant différentes variables. La situation comportementale d'un cas définit des comportements élémentaires (séquences de valeurs ou événements) et des contraintes temporelles pour caractériser une expérience utile et précise au sein d'un enregistrement. Deuxièmement, le modèle de représentation des indices permet la gestion d'historiques hybrides (historiques échantillonnés et à événements) et présente de meilleures possibilités pour la découverte de connaissances. Nous permettons en effet l'extraction structurée de cas grâce à la notion de *patron de cas potentiels*. Les cas potentiels instanciés par un patron peuvent ensuite être sauvegardés sous forme de cas concrets, puis mis à jour suivant les raisonnements. Enfin, l'interprétation des cas concrets est facilitée puisque ces cas référencent toujours les données brutes des enregistrements.

Nous abordons dans ce chapitre le problème de l'indexation des cas. Comme nous l'avons montré dans le chapitre précédent, ce problème est crucial pour le raisonnement à partir de cas puisqu'il a des conséquences sur les phases de recherche, de réutilisation et d'apprentissage. Généralement, l'étude de la tâche à aborder par le système mène à la définition d'un vocabulaire d'indexation, donnant les dimensions d'un cas et leurs domaines de valeurs selon lesquelles il est possible d'exprimer les indices du cas. Dans la majorité des systèmes de raisonnement à partir de cas, des indices instantanés sont utilisés : ils représentent les caractéristiques de l'univers étudié à l'instant où le problème a été posé. Contrairement à ces approches, nous étudions une classe de problèmes dans laquelle la prise en compte de l'évolution de certaines variables est requise pour formuler le problème à résoudre. Il s'agit donc de représenter et de gérer des indices sous forme *d'historiques*.

Dans la première partie, nous menons l'état de l'art des approches abordant l'indexation par historiques au sens large. Ces approches sont issues non seulement du domaine du raisonnement à partir de cas, mais aussi de la reconnaissance d'historiques et des bases de données temporelles. Dans la deuxième partie, nous proposons une démarche originale dans la définition du *modèle d'indexation par situations comportementales*. Notre modèle propose une solution générique pour l'indexation des cas pour la classe de problèmes étudiée. A la fin de cette partie, nous illustrons les notions utilisées sur l'exemple de la nutrition des plantes. L'utilisation concrète de ce modèle est facilitée par notre plate-forme à objets que nous présentons dans les chapitres III et IV. Le chapitre V donnera enfin une application complète de notre modèle d'indexation dans la réalisation de notre prototype d'assistant à la navigation sur le Web.

1 Approches existantes d'indexation par historiques

Dans le cadre de l'état de l'art sur l'indexation de connaissances par historiques, nous regroupons les approches existantes en deux groupes : d'une part les approches en RàPC et d'autre part celles liées à la reconnaissance d'historiques et aux systèmes de gestion de bases de données temporelles.

1.1 Approches basées sur le raisonnement à partir de cas

Peu de systèmes de RàPC ont abordé l'indexation par historiques qui nécessite la prise en compte d'une dimension temporelle dans les cas. Toutefois, les systèmes de raisonnement à partir de cas donnant une dimension temporelle à leurs cas n'ont pas tous recours à une indexation par historiques. Par exemple, dans certaines approches (Brandau *et al.*, 1991 ; Kopeikina *et al.*, 1988 ; Deter, 1994) l'observation d'un réseau de communication fait partie inhérente de l'application mais les historiques des observations n'ont pas été jugés pertinents pour l'indexation, bien que la dimension temporelle ait été prise en compte dans la description incrémentale du problème (Deter, 1994) ou dans la représentation de plans et de comportements attendus dans la solution des cas (Brandau *et al.*, 1991 ; Kopeikina *et al.*, 1988). Ainsi, nous présentons uniquement les approches dans lesquelles les historiques sont des indices, c'est-à-dire qu'ils déterminent dans quel cadre un cas est utile et applicable. Nous montrons alors les limites de ces systèmes dans l'objectif de la définition d'un modèle général d'indexation par historiques, adapté à la classe de problèmes visée.

1.1.1 Présentation des systèmes existants

Dans un premier temps, nous présentons plus en détail le système REBECAS (Rougegrez, 1994a) qui aborde la prévision de l'évolution des incendies de forêt et le système SINS (*Self-Improving Navigation System*, Ram & Santamaria, 1997) qui a été conçu dans le cadre de la commande d'un

robot mobile. Ces deux systèmes abordent des problèmes complexes et apportent des solutions complètes et originales. Nous présenterons ensuite brièvement le contexte et les objectifs des autres systèmes et approches ayant utilisé des historiques dans l'indexation des cas.

1.1.1.1 Prévion de l'évolution des feux de forêts : le système REBECAS

Le but du système REBECAS (Rougegrez, 1994a ; Rougegrez, 1994b) est de prédire l'évolution d'un processus complexe et en particulier de prédire l'évolution des incendies de forêt. Le domaine des feux de forêt est complexe car les paramètres agissant sur l'évolution du feu sont nombreux et leurs effets sont mal connus. De plus, la plupart des modèles de feux de forêt ne prennent pas en compte les évolutions de l'environnement (du vent par exemple). Le système REBECAS permet de prendre en compte ces évolutions sous la forme d'historiques d'événements pour essayer de donner une meilleure prévision. L'observation du feu se fait grâce à des relevés effectués sur le front du feu et à l'utilisation de cartes pour connaître le relief et la végétation parcourue. Les relevés ne sont pas systématiques, il peut donc y avoir une incertitude sur l'évolution entre deux relevés successifs et il n'est pas possible de savoir si un changement a été brusque ou progressif.

Dans REBECAS, un cas représente l'ensemble des données d'observation disponibles pour le déroulement complet d'un feu. Un cas est décomposé suivant plusieurs directions de propagation du feu. Pour chaque direction, un cas comprend les historiques (spatio-temporels) des paramètres pris en compte (relief, vent, végétation) et de la description de la propagation depuis son point de départ jusqu'à sa fin (cf. Figure II-1). Pour une direction donnée, un ensemble d'événements est généré pour chaque variable observée :

- *Relief* : un événement est un point donnant l'altitude et la distance au prochain point.
- *Végétation* : un événement donne le type de végétation (broussailles, conifères, vignes...) et son étendue (distance).
- *Vent* : un événement daté contient le lieu de la mesure ainsi que la direction et la vitesse du vent.
- *Propagation* : un événement daté est constitué notamment de la distance parcourue depuis le départ du feu.

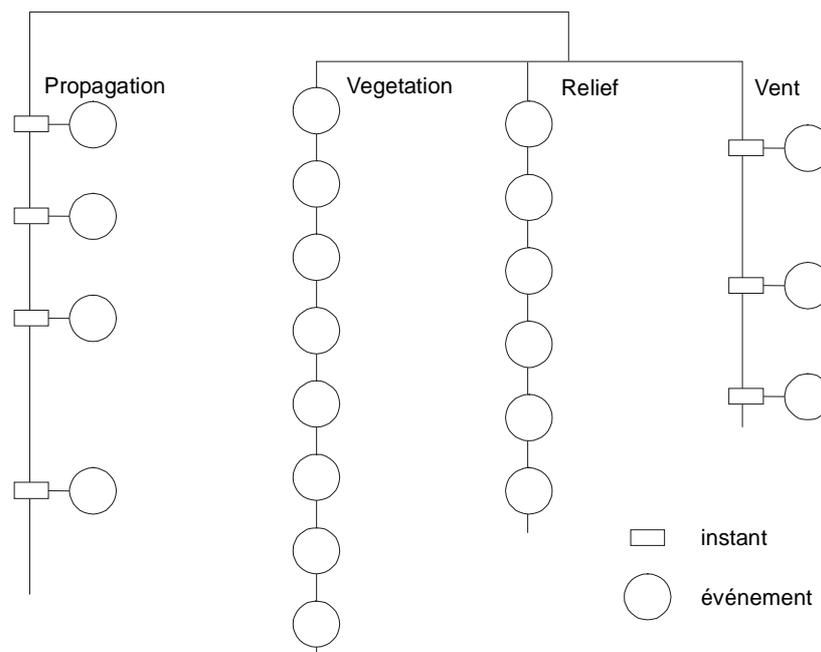


Figure II-1 : historiques d'un cas suivant une direction de propagation

Le système cherche à produire une prévision à chaque nouvelle entrée de données pour la direction de propagation correspondante. Le problème courant est alors constitué des différents historiques de l'évolution des variables depuis le départ du feu. La structure de la mémoire de cas est linéaire, et le but de la phase de recherche est d'identifier un cas qui, suivant une direction de propagation, présente les historiques les plus similaires à ceux du feu courant. Toutefois, selon l'auteur, la phase de recherche ne peut pas prendre en compte tous les paramètres simultanément car, d'une part il n'est pas possible d'agrèger les comparaisons des variables de nature différente, et d'autre part la recherche serait trop restrictive pour retrouver une situation similaire. Ainsi, la recherche s'effectue d'abord selon chaque *point de vue* c'est-à-dire selon un paramètre (vent, végétation et relief). Les meilleures séquences identifiées dans chaque cas et dans chaque direction, suivant tous les points de vue, sont alors retournées. Puis, le meilleur cas est sélectionné grâce à une mesure cherchant notamment à minimiser le décalage temporel entre ces séquences (cf. Figure II-2).

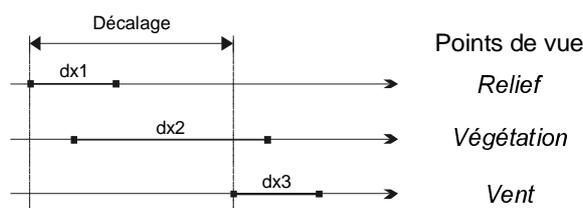


Figure II-2 : choix du cas minimisant le décalage

Pour la mise en correspondance des historiques, des algorithmes de comparaisons approximatives de chaînes de caractères sont utilisés après des étapes de transformation des événements. En effet, pour rendre cette comparaison possible et plus significative, une étape d'interprétation et une étape d'adéquation sont effectuées. L'étape d'interprétation du relief consiste, par exemple, à transformer les événements dans leur forme initiale, en une succession de formes de relief (crête, col, falaise...) grâce à l'utilisation d'un moteur d'inférence. Pour le vent, il s'agit d'exprimer les variations entre deux mesures successives plutôt que leurs valeurs absolues. L'étape d'adéquation consiste alors à préparer concrètement la mise en correspondance : vérification des longueurs des séquences à comparer et ajout de mesures hypothétiques pour compléter les séquences d'observation du vent par exemple. Puis la mise en correspondance est effectuée. Pour le relief et la végétation une distance est définie pour les événements, et une méthode de recherche de motifs approximatifs dans une chaîne de caractères est utilisée. Cette méthode permet de sélectionner la meilleure séquence en tenant compte du fait que les événements (pris comme des caractères) peuvent être inversés ou manquants. Pour le vent, le système calcule les distances pour la direction et la force pour chaque événement et agrège ces résultats en calculant leur somme, la plus petite somme étant alors sélectionnée.

Une fois le meilleur cas identifié suivant cet algorithme, la phase d'adaptation consiste d'abord à déterminer l'événement du cas sélectionné qui sera utilisé comme repère, puis les événements suivant ce repère sont transformés de manière à générer des événements de prévision dans le cas cible courant. Cette transformation permet de générer des événements qui gardent la même distance relative et la même durée relative. La phase d'apprentissage de REBECAS est effectuée manuellement par la description de nouveaux cas à partir des dossiers rédigés par les sapeurs pompiers.

1.1.1.2 Commande de robots mobiles : le système SINS

Le but du système SINS (*Self-Improving Navigation System*, Ram & Santamaria, 1993 ; Ram & Santamaria, 1997) est de commander de manière réactive un robot mobile dans un environnement réel. L'architecture générale est présentée dans la Figure II-3. Le module de navigation permet de contrôler le robot suivant des schémas de réaction comme « avancer vers le but », « éviter un obstacle », « mouvement aléatoire ». Le module d'apprentissage et d'adaptation dans lequel le raisonnement à partir de cas est effectué doit déterminer les meilleurs paramètres des schémas, étant

donné l'environnement courant. Contrairement à ACCBARR (Moorman & Ram, 1992), son prédécesseur, SINS prend en compte les historiques de l'évolution des paramètres décrivant une situation et possède un module d'apprentissage automatique.

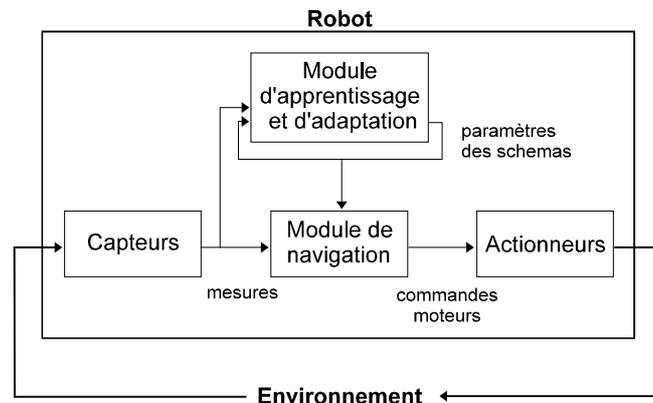


Figure II-3 : architecture de SINS

Dans l'observation du processus nécessaire au raisonnement à partir de cas, deux types de variables peuvent être identifiés : les variables de mesure décrivant l'environnement (par exemple la densité des obstacles et la position par rapport au but), et les variables représentant les paramètres des schémas de réaction. Les valeurs des différentes variables sont connues à des intervalles de temps fixes et le nouveau problème est également construit à intervalles de temps fixes. Un nouveau problème est représenté par les évolutions des variables pour une durée donnée. De manière identique, un cas source représente ces évolutions issues d'expériences passées pour une durée fixée. Basée sur une structure linéaire de la base de cas, la phase de recherche permet d'identifier le cas dont le recouvrement (cf. Figure II-4) avec le problème courant est optimum. Plus précisément, par une mise en correspondance progressive de chaque cas en remontant dans le temps, le meilleur recouvrement est identifié en utilisant une mesure de similarité. Toutes les valeurs étant numériques, la similarité repose sur le calcul des écarts quadratiques sur chaque variable, qui sont ensuite agrégés par une moyenne pondérée.

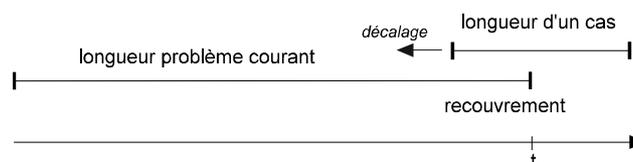


Figure II-4 : principes de la recherche dans SINS

La partie du cas non comprise dans le recouvrement est alors utilisée dans l'adaptation pour déterminer les séquences de réglages (cf. Figure II-5) en tenant compte de la similarité calculée dans la phase de recherche et également d'un facteur aléatoire permettant de modifier les paramètres pour explorer les réactions du système.

La phase d'apprentissage permet finalement d'exécuter deux types d'actions :

- Si le meilleur cas utilisé dans ce raisonnement présente une similarité trop éloignée du cas cible courant, SINS enregistre le cas cible courant dans sa mémoire. Le nombre de cas de la base peut toutefois être limité pour empêcher la création d'une base de cas trop importante.
- Dans le cas contraire, le cas source est modifié ou complété, sans sauvegarder le cas cible courant. Si les paramètres déterminés par le cas source ont été bénéfiques, le cas source est renforcé et/ou allongé dans le temps sinon il est modifié pour explorer les réactions du robot dans les raisonnements suivants. Dans tous les cas, le cas source devient alors un *cas*

virtuel dans lequel les données ne correspondent plus à de réelles observations puisqu'elles sont modifiées en tenant compte de l'évaluation de la réutilisation du cas source.

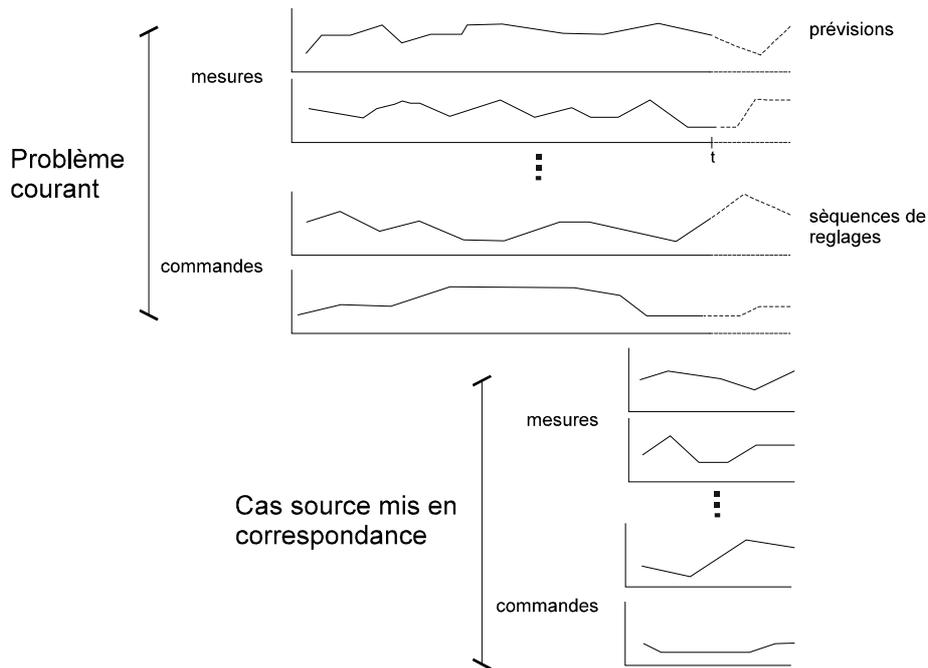


Figure II-5 : mise en correspondance dans SINS

1.1.1.3 Autres systèmes et approches de RàPC

Dans le domaine de la prévision de l'évolution de processus, l'approche de Nakhaeizadeh (Nakhaeizadeh, 1994) a été appliquée dans le domaine économique. Le système de Bull *et al.* (Bull *et al.*, 1997) détecte des risques épidémiologiques et permet également la prévision de l'expansion des épidémies ou de leurs conséquences. Dans le domaine médical, le système ICONS (Schmidt *et al.*, 1996) effectue des estimations de l'évolution de la fonction rénale d'un patient. Le système PADIM « Poste d'Aide à la Décision Immédiate » (Fuchs *et al.*, 1995) aide un opérateur en charge de la supervision d'un procédé complexe en lui proposant les descriptions visuelles les plus adaptées. Enfin, le système RADIX (Corvaisier *et al.*, 1998) est un assistant de navigation sur le Web.

1.1.1.3.1 Prévision économique : l'approche de Nakhaeizadeh

L'approche de Nakhaeizadeh (Nakhaeizadeh, 1993) a été appliquée, dans le domaine économique, à la prévision de taux d'intérêt et à la prévision du nombre d'immatriculations de voitures et de camions en France. Une méthode générale est proposée pour la prévision de l'évolution de processus. Etant donné Y une variable échantillonnée, le système doit prévoir, à l'instant t , l'évolution de Y pour l'instant suivant $t + 1$. En respectant, la structuration d'un cas en (*problème, solution*), un cas source est alors un couple $((Y_{s-k}, \dots, Y_s), Y_{s+1})$ et le cas cible $((Y_{t-k}, \dots, Y_t), Y^*)$ dans lequel Y^* doit être estimée. Le système recherche alors dans la base de cas (ensemble des séquences passées de l'évolution de la variable Y) en les comparant avec la séquence du problème courant avec un algorithme de plus proches voisins. La prévision peut alors être faite par adaptation de la solution de chaque cas identifié (avec une moyenne par exemple). Dans (Nakhaeizadeh, 1994), cette approche est généralisée dans le sens où la partie problème d'un cas est composée de toutes les valeurs ayant un impact significatif sur la variable à estimer : ceci inclut différentes valeurs, y compris les valeurs passées de Y .

1.1.1.3.2 Préviation épidémiologique : l'approche de Bull *et al.*

Le système de Bull *et al* (Bull *et al.*, 1997) permet de détecter des risques épidémiologiques et de prédire l'expansion des épidémies et leurs conséquences. L'entité d'observation est le *scénario*, noté $s \in S$, qui donne l'état de la santé publique et de la charge des services de santé pour une semaine et pour une localité. Une séquence de scénarios successifs est noté $\sigma = s_0 s_1 \dots s_n \in S^*$ avec $s_i \in S$ pour tout i . Une base de cas $\Sigma \in S^*$ est alors en ensemble de séquences de scénarios. Les cas sources n'ont pas d'existence concrète et représentent n'importe quelle sous-séquence de la base : $s\sigma \in \Sigma \Rightarrow \sigma \in \Sigma$ et $\sigma s \in \Sigma \Rightarrow \sigma \in \Sigma$.

Le problème du cas cible est alors exprimé par une séquence particulière, nommée σ . L'objectif du raisonnement est d'identifier la séquence la plus similaire σ' dont la longueur est plus grande que celle de σ .

$$\begin{array}{c} \sigma' = u_0 u_1 \dots u_m u_{m+1} \dots u_{m+k} \\ \downarrow \\ \sigma = s_0 s_1 \dots s_n \dots \end{array}$$

La séquence solution ($u_{m+1} \dots u_{m+k}$) est utilisée par adaptation suivant l'objectif du raisonnement (détection ou préviation). L'approche générale reste très proche de celle de Nakhaeizadeh.

1.1.1.3.3 Préviation médicale : le système ICONS

Dans le domaine médical, le système ICONS (Schmidt *et al.*, 1996) permet de prévoir l'évolution de la fonction rénale d'un patient. Chaque jour, plusieurs dizaines de paramètres sont mesurées ou calculées pour représenter l'état détaillé d'un patient. Ces informations sont ensuite traitées pour obtenir une séquence de symboles donnant les états des sept derniers jours pour former le problème courant, comme par exemple la séquence suivante :

(selective tubular damage, reduced kidney function, reduced kidney function, selective tubular damage, reduced kidney function, reduced kidney function, sharply reduced kidney function).

L'objectif du système est alors de prévoir la suite de l'évolution. Cette représentation de base est alors analysée pour générer la représentation de l'état du patient suivant quatre *tendances* sur des horizons temporels différents : état courant, état suivant les deux derniers jours, état sur les sept derniers jours, et l'état moyen sur les sept derniers jours. Un cas source comprend une description de structure identique et donne comme solution, l'évolution effective des trois jours suivants. Le raisonnement effectue alors une phase de recherche suivant les neuf états possibles de l'évolution du patient. Pour chaque état possible, le cas le plus similaire suivant les différentes tendances est retourné. Les cas retrouvés sont alors présentés à l'utilisateur suivant leur degré de similarité, et donnent une évolution possible de l'état du patient. L'apprentissage est effectué par l'ajout de cas dans la base, et l'organisation de la mémoire qui repose sur une hiérarchie de prototypes est mise à jour.

1.1.1.3.4 Aide à la supervision de procédés : le système PADIM

Le système PADIM (Poste d'Aide à la Décision Immédiate, Fuchs *et al.*, 1995) aide un opérateur en charge de la supervision d'un procédé complexe en lui proposant les descriptions visuelles les plus adaptées. Dans un procédé industriel complexe, il existe de multiples façons de visualiser les informations de supervision, et comme il est impossible de tout visualiser, la sélection des

informations pertinentes est un problème important. Cette sélection est effectuée par PADIM en réutilisant des cas sources indexés par une situation composée :

- d'une partie instantanée donnant notamment l'état du système, le profil de l'utilisateur et le but de la supervision,
- et d'une partie comportementale regroupant les événements pertinents issus de l'évolution du procédé et des actions de l'opérateur.

Un cas source associé à cette situation un environnement de supervision (éléments graphiques) comme solution. Etant donné la situation courante, PADIM commence d'abord par rechercher le cas le plus similaire en effectuant un filtrage sur la partie instantanée puis une sélection en prenant en compte la partie comportementale (séquence d'événements). La mesure de similarité pour comparer deux séquences suppose que les événements de la séquence du cas cible se retrouvent tous et dans le même ordre dans la séquence d'un cas source. Cette mesure prend en compte la représentativité et la dispersion relative de deux séquences à comparer (cf. Figure II-6).

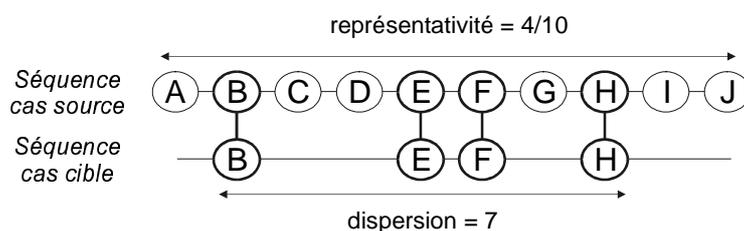


Figure II-6 : mise en correspondance de deux séquences dans PADIM

Suite à la phase de recherche, la disposition et le choix des éléments graphiques à utiliser sont déterminés par l'adaptation du cas source identifié, notamment en prenant en compte les résultats de la mise en correspondance et des connaissances spécifiques sur les différents éléments. L'utilisateur a alors la possibilité de réviser la solution proposée en la modifiant. Enfin, la phase d'apprentissage, contrôlée par un expert, permet d'enrichir la base de cas avec cette nouvelle expérience de supervision.

1.1.1.3.5 Assistance à la navigation : le système RADIX

Le système RADIX (Corvaisier *et al.*, 1998 ; Corvaisier *et al.*, 1997) est un assistant à la navigation sur le Web en cours de conception et n'intègre pas encore toutes les phases du raisonnement à partir de cas. Cependant, ce système aborde plus particulièrement les problèmes de représentation des sessions de navigation et de définition des mesures de similarité à utiliser pour la phase de recherche. Un cas représente une session de navigation qui est composée d'un ensemble ordonné de tentatives unitaires de recherche (TU). Chaque TU se décompose à son tour en une séquence d'unité de recherche (UR) reliant deux pages Web par une transition. Au sein d'une session, les changements de TU sont détectés par des actions spécifiques comme la sélection d'un signet, le choix d'un lien dans l'historique ou encore l'édition manuelle d'une adresse. La phase de recherche va permettre d'identifier des sessions qui présentent une TU similaire à la TU courante de l'utilisateur. Cette recherche commence par un filtrage sur la signature des TU formée du vocabulaire (ensemble de mots pertinents issus des pages visitées), du type d'information recherchée et du type de besoin d'information (précis, thématique ou exploratoire). Puis une phase de sélection est effectuée pour prendre en compte de manière plus précise : la séquence ordonnée de pages et de transitions, le type et le besoin d'information et la date de la TU. La mesure de similarité, utilisée pour la séquence de pages et de transitions, est identique au système PADIM (cf. Figure II-6). A terme, le système vise la réutilisation des sessions ainsi retrouvées, pour proposer à l'utilisateur des pages pertinentes à visiter, des libellés de requêtes et des listes de serveurs à interroger.

1.1.2 Limites des systèmes existants

Au regard de notre classe de problèmes et de notre démarche visant à faciliter la mise en place des représentations et des mécanismes liés à l'indexation, nous mettons en évidence trois limitations des approches existantes : l'absence d'approche générale pour l'indexation, l'insuffisance de la gestion des cas pour la découverte de connaissances et l'absence de la gestion d'indices basés sur des observations hybrides.

1.1.2.1 Absence d'approche générale pour l'indexation

Parmi les démarches prises pour l'indexation dans les travaux existants, nous pouvons identifier deux groupes :

- les approches spécifiques à un problème bien défini : ICONS pour la prévision de l'évolution de la fonction rénale d'un patient, PADIM pour l'aide à la supervision et RADIX pour l'assistance à la navigation,
- les approches présentant certains aspects réutilisables pour l'indexation :
 - REBECAS qui est appliqué à la prévision des incendies de forêt mais dont les contraintes de sa généralisation ont été brièvement explicitées (Rougegrez, 1994a, chapitre 6),
 - SINS dont l'approche dépasse la commande de robots pour aborder la classe des problèmes *continus* (Ram & Santamaria, 1997),
 - et enfin les approches de Bull *et al.* et de Nakhaeizadeh dont les principes simples peuvent être réutilisés dans d'autres domaines.

Toutefois, aucune des approches précédentes ne présente un modèle d'indexation générique. C'est à chacun d'extraire de ces travaux, les processus d'indexation généraux touchant la recherche, la réutilisation et l'apprentissage, tout en identifiant leurs conditions d'application. Enfin, aucun outil ni méthode n'est proposé pour mettre en pratique ces processus généraux.

1.1.2.2 Gestion des cas insuffisante pour la découverte de connaissances

La classe de problèmes visée est notamment caractérisée par une nécessité d'explicitement des nouvelles connaissances parmi les nombreuses données collectées. Par exemple, dans la nutrition des plantes, il s'agit de présenter aux utilisateurs des comportements provenant de cas enregistrés pour permettre d'enrichir les connaissances du domaine. La découverte de connaissances à l'aide d'un système de raisonnement à partir de cas est selon nous basée sur trois opérations importantes :

- extraction automatique d'enseignements (cas ou parties de cas) à partir des données brutes,
- mise à jour des enseignements suivant les raisonnements effectués,
- interprétation par des experts du domaine des connaissances issues des enseignements grâce à leurs indices.

Nous montrons dans cette section que les approches existantes ne permettent pas d'effectuer l'ensemble de ces opérations de manière satisfaisante sur le plan de la gestion des cas.

1.1.2.2.1 Extraction automatique d'enseignements à partir des données brutes

L'extraction automatique d'enseignements permet durant un raisonnement de préciser un ensemble d'indices effectifs pour une solution proposée. Les systèmes ICONS et PADIM ne permettent pas cette extraction et utilisent des indices effectifs figés avant le raisonnement. Par exemple, dans PADIM, l'historique des événements liés au système et à l'opérateur est explicitement associé au cas de supervision. Dans le système ICONS, l'expérience représentée par un cas contient une description comportementale donnant l'évolution de l'état du patient dans les sept jours précédents. Ces systèmes

ne permettent donc pas d'extraire automatiquement des comportements pertinents en fonction du raisonnement courant. Il se borne à réutiliser directement les comportements décrits dans les cas existants et explicites. De par le manque de connaissances et le nombre important de données brutes enregistrées, nous supposons qu'il n'est pas possible, dans le cadre de notre classe de problèmes, d'identifier de manière exhaustive tous les enseignements disponibles.

D'un autre côté, les cinq autres systèmes permettent cette extraction suivant deux méthodes différentes. Premièrement, dans SINS, REBECAS, et RADIX, un cas source sera réutilisé si une partie *variable* de sa description correspond au problème courant. Par exemple, dans REBECAS, un cas représente l'évolution d'un feu tout entier depuis son début jusqu'à sa fin. Un cas source est alors séparé dynamiquement en une partie qui est mise en correspondance (les indices effectifs) et une partie solution qui donnera par adaptation une prévision de l'évolution de l'incendie. De même, dans SINS, un cas est constitué des évolutions des attributs de l'environnement et des paramètres de commande pour une durée déterminée. Suivant la mise en correspondance, le cas source est également séparé dynamiquement en deux parties comme dans REBECAS (cf. Figure II-7). Deuxièmement, dans les approches de Bull *et al.* et de Nakhaeizadeh, un cas n'a pas d'existence propre au préalable, il est identifié dynamiquement avec ses indices effectifs au sein de la base de cas suivant le problème courant.

En définitive, nous pouvons définir deux types de situations d'un cas : les situations *non explicites* et les situations *explicites*. Une situation est dite non explicite lorsque les indices effectifs qui la composent doivent être identifiés dynamiquement pendant la phase de recherche, soit parmi les indices potentiels d'un cas, soit par l'identification d'un nouveau cas. Une situation est dite explicite lorsque ses indices effectifs sont connus avant le raisonnement.

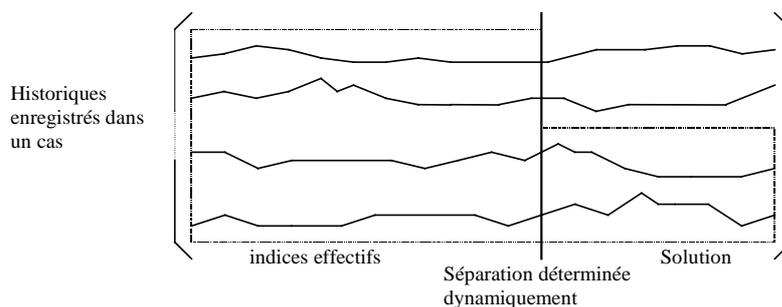


Figure II-7 : exemple de cas avec une situation non explicite dans SINS

1.1.2.2.2 Mise à jour des enseignements

A la suite de l'extraction des ces enseignements et à l'exécution d'un raisonnement qui a mené à un échec ou à un succès, la phase d'apprentissage doit pouvoir modifier les enseignements et leur organisation pour refléter ces résultats. Par exemple, l'échec du raisonnement peut provenir de la prise en compte d'indices effectifs non pertinents ou, au contraire, est révélateur du manque d'indices effectifs pertinents. Au niveau de la gestion des indices, il apparaît donc important de pouvoir modifier la sélection des indices effectifs parmi les indices potentiels d'un enseignement. Il est aussi important de pouvoir associer des indications sur la qualité de cet enseignement (nombres d'échec et de succès par exemple). Ces opérations sont primordiales pour mettre en évidence de nouvelles connaissances valides. Nous distinguons d'une part les systèmes qui n'effectuent aucune mise à jour, et d'autre part SINS qui opère des modifications.

Les systèmes REBECAS et RADIX ainsi que les approches de Bull *et al.* et de Nakhaeizadeh ne font aucune mise à jour des enseignements extraits et utilisés. Ce constat est très négatif sur la qualité et l'efficacité du raisonnement. Tout d'abord, la qualité du raisonnement progresse peu puisque

l'apprentissage se limite à l'ajout d'un nouveau cas. L'identification des enseignements qui auront mené à un échec ou à un succès ne sera pas prise en compte pour les raisonnements futurs. Ainsi, le même échec pourra se reproduire et un succès ne sera pas renforcé. Ensuite sur le plan de l'efficacité, l'identification des enseignements est souvent coûteuse, et si aucune information n'est retenue, il faudra exécuter à chaque fois les mêmes opérations. Par exemple dans REBECAS, l'extraction des enseignements dans les incendies passés et leurs utilisations dans le raisonnement ne mènent à aucune modification. Il n'est donc pas possible de mémoriser si une prévision fut un succès ou si les hypothèses de mise en correspondance étaient justifiées. En définitive, il n'est pas possible de mettre à jour ni les indices effectifs d'un enseignement, ni sa qualité. De plus, cette extraction est coûteuse car il s'agit de mettre en correspondance l'évolution du feu courant sur plusieurs paramètres (relief, vent et végétation). Ces opérations seront toujours à effectuer même pour un problème très similaire.

Seul SINS effectue une mise à jour à chaque raisonnement en modifiant les cas (cas virtuels). Ainsi, il est capable de mettre en évidence et de faire converger les cas vers des comportements plus généraux du processus étudié.

1.1.2.2.3 Interprétation des enseignements par analyse des indices

Après les deux premières opérations, l'interprétation des enseignements par des experts est nécessaire pour enrichir les connaissances du domaine. Le rôle des indices d'un enseignement est alors central dans cette interprétation. Les indices permettent en effet de définir le plus précisément possible le cadre d'application d'un enseignement, c'est-à-dire les conditions d'application de la connaissance que cet enseignement porte sur le processus observé.

SINS présente sur ce point des limitations. En effet, les mécanismes de modification de l'indexation sont basés sur des calculs numériques qui vont permettre de modifier le cas lui-même. Une partie du caractère explicatif des indices est alors perdue puisqu'il ne s'agit plus d'une expérience directement vécue. Une telle méthode pourra poser des problèmes d'intégration dans une application interactive par exemple. L'auteur du système précise lui-même que ces mécanismes rendent difficile une interprétation de plus haut niveau. Toutefois, cette interprétation n'était pas un objectif premier de SINS, le système cherchant en premier lieu l'efficacité du raisonnement.

1.1.2.2.4 Synthèse

	<i>Extraction</i>	<i>Mise à jour</i>	<i>Interprétation</i>
REBECAS (Rougegrez, 1994a)	✓	-	n/a
SINS (Ram & Santamaria, 1997)	✓	✓	-
Nakhaeizadeh (Nakhaeizadeh, 1994)	✓	-	n/a
Bull <i>et al.</i> (Bull <i>et al.</i> , 1997)	✓	-	n/a
ICONS (Schmidt <i>et al.</i> , 1996)	-	n/a	n/a
PADIM (Fuchs <i>et al.</i> , 1995)	-	n/a	n/a
RADIX (Corvaisier <i>et al.</i> , 1998)	✓	-	n/a

Tableau II-1 : gestion des cas et découverte de connaissances dans les approches existantes¹⁹

¹⁹ La notation n/a indique qu'il n'est pas possible de comparer un système puisqu'il ne permet pas l'opération précédente.

Le Tableau II-1 donne la synthèse de notre analyse et répond aux questions suivantes pour chacun des travaux existants :

- L'approche permet-elle d'extraire automatiquement des enseignements ?
- Si oui, sont-ils mis à jour suivant les raisonnements ?
- Si des mises à jour sont effectuées, les enseignements gardent-ils leur caractère explicatif pour une interprétation ?

Suite à notre analyse, nous concluons qu'aucun des systèmes existants ne permet de manière satisfaisante la gestion des cas pour la découverte de connaissances.

1.1.2.3 Absence d'indices basés sur des observations hybrides

Deux types d'historiques peuvent être identifiés dans les travaux existants : les historiques *échantillonnés* et les historiques à *événements*. La majorité des systèmes ne gère que des historiques échantillonnés, c'est-à-dire que les valeurs de l'historique sont obtenues à une fréquence fixe. Par exemple, dans le système de Bull *et al*, les indices sont constitués d'une séquence de scénarios où chaque scénario donne l'état de la santé publique et de la charge des services de santé, pour chaque semaine. L'observation du système ICONS est basée sur plusieurs dizaines de paramètres qui sont traités pour obtenir l'état journalier d'un patient. Seuls PADIM, RADIX et REBECAS gèrent des historiques à événements, où un événement est un changement d'état ou une action considérée comme instantanée. PADIM intègre, dans la description de la situation, des séquences d'événements qui sont issus de l'évolution du procédé et des actions de l'opérateur. RADIX prend en compte les événements de navigations (changement de page). Dans REBECAS, le feu est observé grâce à des relevés effectués sur le front du feu, et les informations continues provenant de cartes sont transformées en historiques à événements.

Toutefois aucun système ne propose un modèle de représentation des indices *intégrant à la fois des historiques échantillonnés et à événements*. Cette possibilité est utile dans le cadre de la classe de problèmes visée pour deux raisons. Premièrement, le processus observé peut être hybride c'est-à-dire qu'il présente dans son fonctionnement à la fois des variables échantillonnées (humidité, rayonnement solaire) et des variables à événements (déclenchement du chauffage ou alarmes). Deuxièmement, dans le cadre d'un système de RàPC, il est important de représenter le comportement du processus de manière pertinente pour le raisonnement. Ainsi, même si toutes les variables de base sont échantillonnées, il peut être utile de transformer ces observations en événements de plus haut niveau pour former de meilleurs indices. Cependant toutes les mesures échantillonnées ne peuvent pas toujours être transformées en événements car cette transformation nécessite des connaissances spécifiques. Par exemple, REBECAS utilise un moteur d'inférence pour transformer des relevés de cartes topographiques en formes de reliefs, considérés par la suite comme des événements. L'absence de ces connaissances peut donc mener à l'étude de variables uniquement échantillonnées ou hybrides.

1.2 Autres approches

En dehors des approches de raisonnement à partir de cas, nous avons étudié deux autres groupes de travaux qui abordent l'indexation par historiques, au sens large, de connaissances ou de données : les approches de reconnaissance d'historique et les systèmes de gestion de bases de données temporelles. Nous présentons tout d'abord ces deux types d'approches puis nous faisons une synthèse montrant que leur problématique sur l'indexation est restreinte par rapport à celle nécessaire pour un cycle de raisonnement à partir de cas.

1.2.1 Reconnaissance d'historiques

Les approches de reconnaissance d'historiques mettent l'accent sur la mise en correspondance d'un historique source et d'un historique cible, et nous les classons en trois types :

- approches de traitement du signal qui opèrent sur des historiques purement numériques,
- approches syntaxiques qui utilisent des historiques formés de séquences de symboles,
- approches à base de scénarios temporels qui mettent l'accent sur la gestion de contraintes temporelles.

1.2.1.1 Approches de traitement du signal

Dans ce type d'approches, un historique est représenté par les valeurs successives prises par une variable numérique échantillonnée, et nous parlons alors de *signal numérique*. Ce signal est généralement issu d'un signal analogique qui aura été préalablement numérisé. Le problème de la reconnaissance d'historiques se pose alors dans ces termes : étant donné un signal cible, il s'agit de reconnaître son évolution par rapport à un ou plusieurs signaux ou modèles de signaux de référence (sources). De multiples approches sont proposées en fonction des propriétés du signal et des objectifs de la mise en correspondance. Une synthèse des fonctions de comparaison existantes est présentée dans (Bassaville, 1988).

Nous donnons dans la suite l'exemple d'un système qui a été appliqué à la reconnaissance d'un signal transitoire rapide (Charles, 1975). Soit le signal cible $X = (x_1, \dots, x_I, \dots, x_M)$ et un ensemble de signaux de référence S dont les éléments sont notés $Y_i = (y_1, \dots, y_I, \dots, y_N)$ avec $M \geq N$. L'objectif du système est de reconnaître dans X , l'un des signaux de référence, sachant que les signaux de référence peuvent se reproduire dans X avec un facteur de proportionnalité et un décalage temporel quelconques. La reconnaissance s'effectue alors en trois étapes :

1. *Prétraitement*. Le maximum de X (position et valeur) et sa polarité sont déterminés.
2. *Sélection grossière*. Tout d'abord, seuls les signaux Y_i ayant la même polarité que X sont retenus. Puis, X est positionné avec chaque Y_i de manière à ce que leurs maximums respectifs coïncident. Le calcul du carré du coefficient de corrélation, noté ρ_i^2 , entre chaque couple de signaux est alors effectué :

$$\rho_i^2 = \frac{\sum_{I=1}^N x_I \cdot y_I}{\sum_{I=1}^N x_I^2 \sum_{I=1}^N y_I^2}$$

Ce coefficient est compris entre 0 et 1, et est égal à 1 lorsque deux signaux sont identiques, (avec un facteur de proportionnalité près). Enfin, seuls les signaux pour lesquels ce coefficient est supérieur à un seuil (0,9) sont gardés pour l'étape suivante de la reconnaissance. Si aucun coefficient est supérieur à ce seuil, la reconnaissance a échoué.

3. *Sélection définitive*. Il se peut que la mise en correspondance basée sur le maximum de chaque signal soit imprécise. C'est pourquoi, dans cette étape, chaque signal est déplacé autour de ce maximum. Le coefficient de corrélation est calculé pour chaque décalage J :

$$\rho_{iJ}^2 = \frac{\sum_{I=1}^N x_{I+J} \cdot y_I}{\sum_{I=1}^N x_{I+J}^2 \sum_{I=1}^N y_I^2}$$

Dans l'expérimentation du système, J varie entre -6 et +6. Le plus grand de ces coefficients est alors associé à chaque signal restant. Enfin, le signal de référence dont le coefficient de corrélation est le plus grand, est donné comme résultat de la reconnaissance.

1.2.1.2 Approches syntaxiques

Dans ce type d'approches, un historique est représenté par une chaîne de caractères issus d'un alphabet. Dans cette modélisation, chaque caractère représente soit un événement soit une mesure du processus observé ou de son environnement. Le problème de la reconnaissance d'un historique revient alors à mettre en correspondance une chaîne de caractères source ou un type de chaîne de caractères, avec une chaîne de caractères cible.

Dans un premier temps, les données brutes, qu'elles soient numériques ou symboliques, issues d'échantillonnage ou de l'occurrence d'un événement, sont traitées pour former la chaîne de caractères cible. Par exemple, les données numériques échantillonnées issues de l'observation d'un électrocardiogramme humain peuvent être segmentées et transformées en une succession de caractères (cf. Figure II-8). De manière similaire, le système Patterns (Morrill, 1998) permet d'exprimer des primitives (signal constant, dépassement de seuils, signal concave ou convexe) et de les relier par des opérateurs (conjonction, disjonction, boucle, espacement) pour former un patron d'évolution à reconnaître.

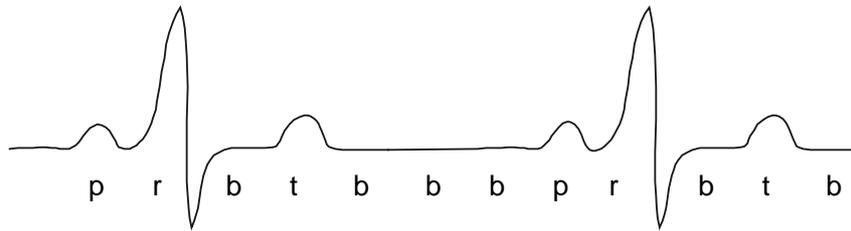


Figure II-8 : électrocardiogramme humain normal (Gonzalez, 1978)

Puis, pour la mise en correspondance, de nombreuses techniques sont applicables : mesure de similarité entre chaîne de caractères (Hall & Dowling, 1980) ou techniques issues de la théorie des langages formels (Gonzalez, 1978). Afin de présenter un exemple d'utilisation de cette dernière approche, nous rappelons brièvement les notions utilisées par cette théorie (alphabet, langage, grammaire et automates).

Un *alphabet*, noté Σ , est un ensemble fini de symboles. Sur cet alphabet, un *mot* ou *phrase* est une succession finie de symboles. Un mot a une longueur qui correspond au nombre de symboles qui le constituent, et ϵ est le mot vide. Si u et v sont deux mots, la concaténation de u et v , notée uv , est un opérateur qui forme un nouveau mot en juxtaposant les symboles de u puis les symboles de v . Σ^* est l'ensemble des mots pouvant être formés à partir de l'alphabet Σ , y compris le mot vide ϵ . Σ^* est un ensemble dénombrable et infini. Par exemple si $\Sigma=\{a,b\}$, a , b , ab , $abba$ sont des mots sur Σ . Un *langage* L est un ensemble de mots ou phrases sur un alphabet Σ , c'est un sous-ensemble de Σ^* . En reprenant l'exemple précédent, un langage peut être défini en extension par l'ensemble de mots $\{abba, baab, abba, baaab\}$. Une *grammaire* $G=(N,\Sigma,P,S)$ définit les règles de syntaxe ou axiomes qui permettent de générer un ensemble de mots avec : N un ensemble fini de symboles non terminaux, Σ l'alphabet (symboles terminaux), P l'ensemble des axiomes, S élément de N le symbole de départ.

Le langage défini par une grammaire G est noté $L(G)$. Plusieurs types de grammaires existent suivant les restrictions sur les axiomes utilisés : grammaire sans restriction (type 0), grammaire contexte-lié (type 1), grammaire contexte libre (type 2), et grammaire régulière (type 3). Nous allons nous intéresser dans cette présentation uniquement aux grammaires régulières. Dans le cas d'une grammaire régulière à droite, les axiomes sont de la forme $A \rightarrow aB$ ou $A \rightarrow a$ où A et B sont éléments de N , et a est élément de Σ^* . Une fois la grammaire et le langage ainsi définis, le problème pratique

de la reconnaissance des mots d'un langage se pose. Suivant le type de la grammaire, une machine de reconnaissance particulière est utilisée :

- Type 0 : machine de Turing.
- Type 1 : automate linéaire (*Linear-bounded automaton*).
- Type 2 : automate à pile (*Push-down automaton*).
- Type 3 : automate fini (*Finite automaton*).

Ainsi pour le langage régulier généré par une grammaire régulière, nous pouvons utiliser un automate fini A que l'on définit par $A=(Q,\Sigma,\delta,q_0,F)$ avec : Q un ensemble d'états, Σ un alphabet, δ une fonction de transitions de $\Sigma \times Q$ dans Q, q_0 élément de Q l'état de départ, et F le sous-ensemble de Q des états finaux. Un mot $w \in \Sigma^*$ est accepté (reconnu) par un automate fini déterministe si et seulement si, après avoir lu le mot, l'automate se trouve dans un état final.

Prenons l'exemple introduit dans la Figure II-8 dans lequel il s'agit de reconnaître un historique normal et un historique anormal. A chaque symbole de l'alphabet $\Sigma=\{p,r,t,b\}$ est associée une évolution primitive de l'électrocardiogramme et la grammaire définit les juxtapositions valides de ces primitives. La grammaire régulière utilisée est $G=(\{S,A,B,C,D,E,H\},\Sigma,P,S)$ où P est défini par les axiomes suivants :

$S \rightarrow p A$	$A \rightarrow r B$	$B \rightarrow b C$
$C \rightarrow t D$	$D \rightarrow b$	$D \rightarrow b E$
$E \rightarrow b$	$E \rightarrow b H$	$E \rightarrow p A$
$H \rightarrow b$	$H \rightarrow b S$	$H \rightarrow p A$

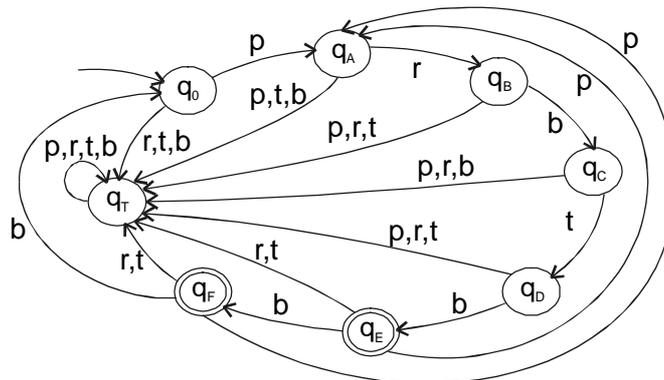


Figure II-9 : automate fini pour la reconnaissance d'un électrocardiogramme normal (Gonzalez, 1978)

L'automate fini permettant de reconnaître le langage $L(G)$, c'est-à-dire l'évolution normale de l'électrocardiogramme, est donné dans la Figure II-9. L'état initial est q_0 , les états finaux sont q_E et q_F c'est-à-dire que si l'automate est dans l'un de ces états, une évolution normale a été reconnue. En revanche, l'état q_T est un état « poubelle », dans le sens où dès qu'une évolution anormale est constatée, l'automate tombe dans cet état.

Nous avons présenté les techniques de base, mais dans le cadre de la reconnaissance de formes, d'autres méthodes plus complexes peuvent être utilisées comme des automates stochastiques prenant en compte les fréquences d'apparition des événements (i.e. des symboles de l'alphabet) pour donner plus d'importance à certains événements dans la reconnaissance, ou bien des automates permettant de reconnaître des phrases imparfaites (absence, insertion ou différence de symboles) (Gonzalez, 1978).

1.2.1.3 Approches basées sur des scénarios temporels

Dans ce type d'approches, un scénario est essentiellement constitué par un graphe de contraintes temporelles entre des observations qu'il s'agit de mettre en correspondance avec les observations effectives (Fontaine, 1996 ; Mounir-Alaoui, 1990 ; Haimowitz & Kohane, 1993a). Pour illustrer ce type d'approches, nous prenons l'exemple de deux systèmes représentatifs IxTeT (Mounir-Alaoui, 1990 ; Dousson & Ghallab, 1994) et TrenD_x (Haimowitz & Kohane, 1993a ; Haimowitz & Kohane, 1993b).

1.2.1.3.1 Le système IxTeT

Le système IxTeT (Mounir-Alaoui, 1990) est un système de gestion de contraintes temporelles qui a été appliqué dans le cadre de la reconnaissance d'historiques. Plus précisément, ce système a été utilisé dans des problèmes de surveillance d'un environnement bien structuré comme un atelier, un parking ou un aéroport. Puis, basée sur les mêmes principes, l'application du système IxTeT a été étendue à la supervision de processus continus (Dousson & Ghallab, 1994) : supervision de turbines à gaz et filtrage d'alarmes dans une unité pétrochimique. Nous présentons ce système en montrant son fonctionnement sur un exemple concret (Mounir-Alaoui, 1990), dans lequel il s'agit de reconnaître le comportement d'un convoyeur dans une usine automatisée. Ce convoyeur doit entrer dans une pièce, effectuer un chargement puis ressortir (cf. Figure II-10).

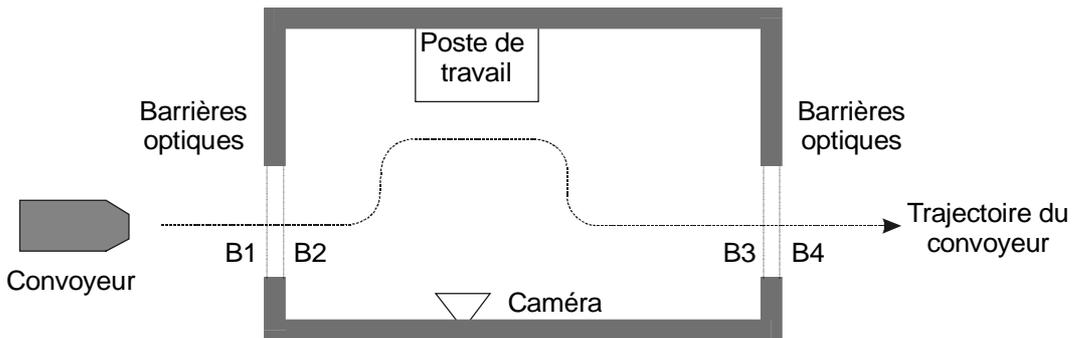


Figure II-10 : déplacement d'un convoyeur dans une usine automatisée (Mounir-Alaoui, 1990)

```
(Situation (S1)
  (Uses
    (fact f1 (on barrier 1) 1)
    (fact f2 (on barrier 2) 1)
    (fact f3 (on inroom 1 conv) 0.8 focus)
    (event e1 (on addweight) 1)
    (fact f4 (on barrier 3) 1)
    (fact f5 (on barrier 4) 1))
  (Such That
    (f1 o f2) (f2 o f3) (e1 o f3) (f4 o f5)
    ((duration f1) < 5 otherwise continue print « Problem at B1 »)
    ((duration f2) < 5 otherwise continue print « Problem at B2 »)
    ((duration f4) < 5 otherwise continue print « Problem at B3 »)
    ((duration f5) < 5 otherwise continue print « Problem at B4 »)
    (20 < (duration f3) < 60 otherwise exit event (on failure))
    ((e1 - (start f2)) < 35 otherwise exit event (on problem))
    (skip f3 if (f1 f2 e1) print « Camera failure »)))
```

Figure II-11 : définition d'une situation dans IxTeT

IxTeT permet de représenter des comportements du convoyeur, appelés situations, en reliant par des contraintes temporelles des faits et des événements. Un fait est une association entre une proposition et un intervalle de temps pendant lequel cette proposition est vraie. Un événement

représente le changement daté de l'état d'une proposition. Une situation est alors décrite dans une syntaxe qui sépare la définition des faits et des événements (clause Uses), des contraintes liant ces faits et événements pour caractériser la situation (clause Such That). La Figure II-11 donne l'exemple d'une situation représentant le comportement normal du convoyeur.

Dans cette situation, les faits f1, f2, f4 et f5 représentent respectivement, le passage du convoyeur au travers des barrières optiques 1,2,3 et 4. Le fait f3 représente la présence du convoyeur dans la pièce qui est détectée par une caméra. Le mot clef focus indique que ce fait est très pertinent dans la reconnaissance de cette situation, et le facteur de 0.8 représente le degré de confiance des observations liées à la caméra. Enfin, l'événement e1 est transmis par un détecteur du poste de travail pour indiquer que le transfert du chargement a été effectué. Ensuite, plusieurs contraintes temporelles sont posées. Premièrement, cette situation définit des contraintes de chevauchement (relation o pour « *overlaps* ») : par exemple, le franchissement de la barrière 2 (fait f2) doit débiter alors que le franchissement de la barrière 1 (fait f1) n'est pas encore fini. Puis, des contraintes sur les durées des faits imposent que le franchissement des barrières optiques (faits f1, f2, f4 et f5) se déroule en moins de 5 secondes. La présence du convoyeur (fait f3) doit avoir une durée comprise entre 20 et 60 secondes, et le début du franchissement de la deuxième barrière ne doit pas être séparé de plus de 35 secondes du transfert du chargement. Enfin, une clause skip permet de ne pas attendre le déroulement du fait f3, si les faits f1 et f2 ainsi que l'événement e1 se sont produits. Ainsi, la syntaxe utilisée est très flexible, et permet de manipuler des événements aussi bien que des faits, ainsi que des contraintes temporelles symboliques ou numériques.

Ce système permet la définition de plusieurs situations à reconnaître qui sont d'abord compilées pour vérifier la cohérence des contraintes et pour construire des structures de données qui permettront de simplifier le processus de reconnaissance (Ghallab & Mounir-Alaoui, 1989). De plus, même si l'expression des situations utilise à la fois des événements et des faits, le système transforme les faits en deux événements (événement de début et de fin) pour gérer de manière plus efficace uniquement un ensemble d'événements. Le système intègre également plusieurs filtres opérant en amont de la reconnaissance proprement dite : filtres temporels pour assurer que les événements sont triés suivant leur date d'occurrence et non suivant leur ordre d'arrivée, et un moteur d'inférence temps réel permettant de générer des événements additionnels suivant des axiomes du domaine.

La reconnaissance se fait alors de manière incrémentale, au fur et à mesure que les événements se produisent. A l'arrivée d'un nouvel événement, trois cas peuvent se présenter :

- s'il fait partie de la description d'une situation et que son occurrence est compatible avec les contraintes de celle-ci, la reconnaissance se poursuit,
- s'il fait partie de la description d'une situation mais que des contraintes sont violées, la situation en cours de reconnaissance peut éventuellement être rejetée,
- s'il ne fait partie d'aucune description de situation, il est rejeté.

Ainsi, durant cette analyse en continue, des situations sont reconnues et les connaissances qui y sont associées permettent de déclencher une action appropriée (alarme par exemple). Le système est conçu pour pouvoir fonctionner dans le cadre d'une application comportant une douzaine de descriptions de situations et quelques dizaines de types d'événements, avec un temps de réponse inférieur à la seconde.

1.2.1.3.2 Le système Trend_x

Trend_x est un système de reconnaissance d'historiques fondés sur la représentation et la gestion de *patrons de tendance* (*trend templates*) (Haimowitz & Kohane, 1993a ; Haimowitz & Kohane, 1993b). Bien que l'approche soit plus générale, le système a été étudié et évalué dans le domaine médical, et en particulier sur le diagnostic des troubles de croissance chez l'enfant. Une *tendance* est un motif

d'évolution cliniquement pertinent, issu d'une séquence de données observées et ordonnées chronologiquement. De plus, ces tendances peuvent prendre en compte de multiples variables d'observation. Un patron de tendance définit alors un type de tendance auquel est associée une hypothèse de diagnostic. De manière générale, un patron de tendance est composé de quatre parties :

- informations liées à ce patron (type de diagnostic par exemple).
- ensemble de points de repère et d'intervalles sur une échelle temporelle. Ces points de repère sont imprécis et possèdent une tolérance.
- ensemble de contraintes de valeurs portant sur des variables d'observation pour un intervalle particulier. Une contrainte est de la forme :

$$m \leq f(D) \leq M$$

où D est l'ensemble des données d'observation lié à un intervalle, f représente une fonction à valeurs réelles, m et M sont des nombres réels ou sont égaux à l'infini.

- ensemble de liens avec d'autres patrons qui seront activés par l'échec d'une reconnaissance.

La Figure II-12 donne un extrait de la description d'un patron de tendances pour la reconnaissance de l'évolution normale de la croissance chez un enfant mâle. Trois points de repère sont définis : naissance, début de la puberté et arrêt de la croissance. Puis trois intervalles temporels sont définis (Int1, Int2 et Int3) permettant d'identifier des périodes pertinentes pour la description du patron. Sur ces intervalles, différentes contraintes de valeurs sont alors posées. Sur l'intervalle Int1, une première contrainte indique que le poids et la taille doivent varier d'une façon similaire. Puis, sur l'intervalle Int2, les écarts types de vitesse de variation du poids et de la taille doivent rester proche de zéro. Enfin, sur l'intervalle Int3, d'autres contraintes sont posées sur l'âge osseux, les hormones de croissance (GH) et les hormones de la thyroïde (T4). Des liens entre ces contraintes et d'autres patrons permettront d'aiguiller la reconnaissance vers d'autres hypothèses envisageables dans le cas de leur violation. Par exemple, si la contrainte portant sur la taille durant l'intervalle Int2 n'est pas satisfaite, le patron pour diagnostiquer une puberté en retard sera étudié.

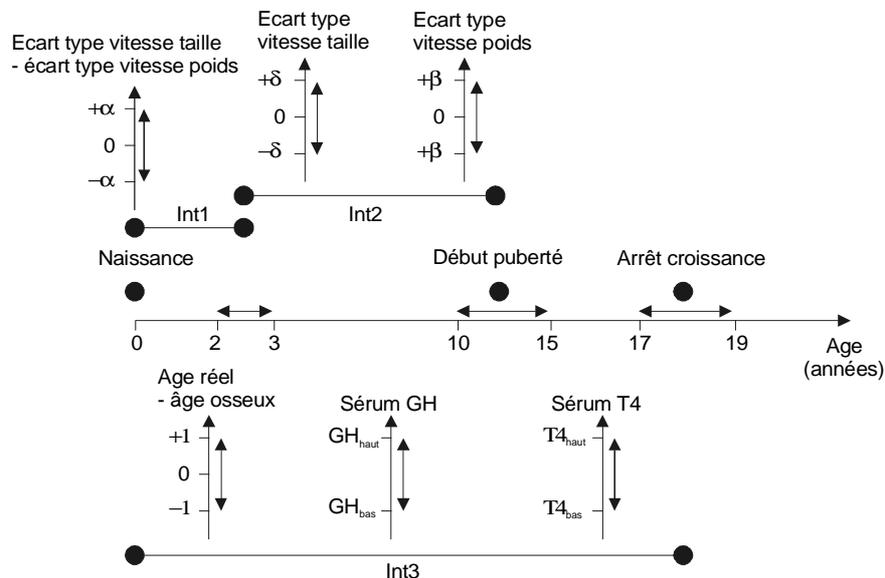


Figure II-12 : extrait d'un patron de tendances (Haimowitz & Kohane, 1993b)

Pratiquement, $TrenD_x$ permet d'établir un diagnostic en mettant en correspondance les données disponibles du patient avec un ensemble de patrons. Le système gère un ensemble d'hypothèses issues de l'instanciation de différents patrons. Chaque patron peut également être instancié plusieurs fois,

s'il y a différentes façons de vérifier les contraintes. L'instanciation de nouvelles hypothèses est guidée par les violations des contraintes d'un patron.

1.2.2 Bases de données temporelles et gestion d'historiques

Les bases de données conventionnelles ont été conçues pour stocker et manipuler les données les plus récentes à propos d'une organisation ou d'une activité : au fur et à mesure que les données sont mises à jour, les anciennes données sont perdues. Ce type de bases de données, appelé bases de données instantanées (*snapshot databases*), ne sont pas adaptées pour la gestion des données liées au passé, au présent et au futur, qui est nécessaire pour la prise en compte d'historiques. Les bases de données temporelles abordent cette problématique. Nous décrivons tout d'abord les axes de recherches sur les bases de données temporelles spécialisées pour la gestion d'historiques (aussi appelés *chroniques*). Nous présentons ensuite un exemple récent d'un de ces systèmes, nommé TEMPOS (Canavaggio, 1997).

1.2.2.1 Systèmes de gestion de bases de chroniques

Les systèmes de gestion de bases de données de chroniques ou TSMS (*Time Series Management System*) forment un domaine particulier au sein des recherches en bases de données temporelles et présentent de nombreuses spécificités (Segev *et al.*, 1995), dont notamment :

- la gestion de la notion de groupe de chroniques,
- l'association de calendriers aux chroniques,
- la nécessité d'opérateurs pour des requêtes d'agrégation et de statistique.

Une analyse précise des besoins pour la gestion de ce type de données est menée dans (Dreyer *et al.*, 1994) et la plupart des systèmes et approches existants pour les bases de données temporelles ne sont pas adaptés à ces spécificités. Seuls les travaux décrits dans (Segev & Shoshani, 1993 ; Segev & Chandra, 1993 ; Canavaggio, 1997) proposent un modèle de chroniques et un ensemble d'opérateurs spécifiques. Dans le premier modèle, décrit dans (Segev & Shoshani, 1993), une chronique ou TS (*Time Sequence*) représente un ensemble ordonné de couples (t,a) , dans lequel t représente le temps et a la valeur d'un attribut. Un TSC (*Time Series Collection*) regroupe un ensemble de chroniques homogènes (par exemple les historiques des salaires des employés d'une société). Un ensemble d'opérateurs est alors défini pour la manipulation des TSC : sélection, agrégation, accumulation, restriction et composition.

Ce modèle a été étendu par la suite (Segev & Chandra, 1993) notamment avec la notion de *concept*. Dans ce modèle, une chronique est associée à un ensemble d'informations qui peut être étendu par l'utilisateur :

- le nom de la chronique,
- le calendrier définissant un ensemble d'instantanés pour lesquels la chronique doit avoir une valeur,
- un ensemble d'instantanés d'exception issus du calendrier pour lesquels la chronique ne peut fournir de valeur,
- la période (*life span*) donnant les instantanés de début et de fin de la chronique,
- le mode de mise à jour de la chronique (chronique dérivée d'autres chroniques ou issue de données brutes),
- la fréquence de mise à jour (par exemple une fois par mois),
- le type de la chronique (cf. Figure II-13) : en *escalier* (la dernière valeur est valide jusqu'à la prochaine mesure), *discrète* (entre deux mesures, la valeur ne peut être calculée ou n'a pas de sens), *continue* (une fonction d'interpolation est utilisée pour obtenir la valeur de la chronique entre deux mesures), ou alors définie par l'utilisateur.

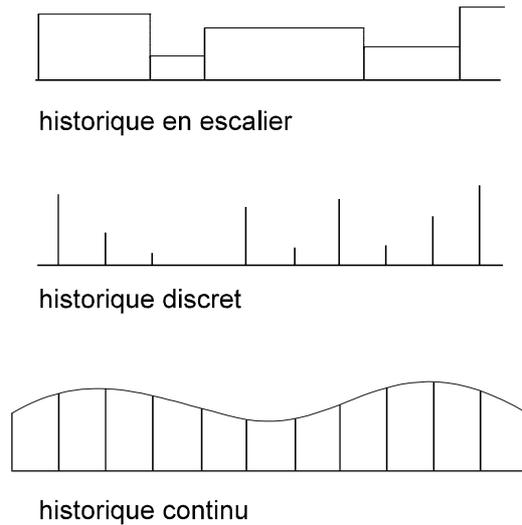


Figure II-13 : types de chroniques (Segev & Shoshani, 1993)

La notion de *concept* permet alors de grouper des chroniques non homogènes et de créer des *concepts complexes* au travers d'une hiérarchie d'héritage. De plus, une même chronique peut appartenir à plusieurs *concepts*. Des opérateurs spécifiques aux chroniques sont également proposés : moyenne mobile, auto-régression, agrégation (par exemple minimum, maximum et moyenne sur une période et/ou suivant une propriété), conversion (par exemple passage d'une chronique de fréquence journalière à une fréquence mensuelle).

1.2.2.2 Le système TEMPOS

TEMPOS « *Temporal Extension Model for Persistent Object Servers* » est un système basé sur le SGBD O₂, qui permet notamment la gestion multigranulaire des valeurs temporelles et la manipulation d'historiques (Fauvet *et al.*, 1997 ; Canavaggio, 1997). Comme dans la plupart des systèmes d'information, TEMPOS est basé sur un modèle granulaire du temps : le temps est représenté par la droite des réels qui est discrétisée par une partition en une suite d'intervalles consécutifs disjoints. Le *chronon* est l'unité la plus petite, c'est un intervalle de temps non décomposable et de durée fixe. D'autres unités peuvent alors être définies par une partition de l'ensemble des chronons. Dans TEMPOS, l'ensemble des unités est muni d'une relation d'ordre partiel « est plus fine que » (cf. Figure II-14) et des opérateurs de conversion et de manipulation ont été définis. Les différentes unités peuvent alors être utilisées dans les requêtes et dans la représentation des historiques.

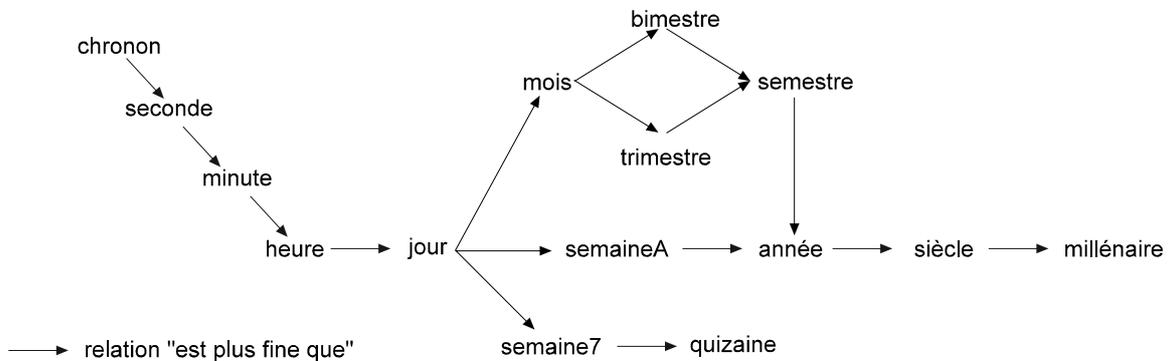


Figure II-14 : différentes unités de temps dans TEMPOS (Canavaggio, 1997, page 48)

Dans TEMPOS, un *instantané* définit un couple $\langle M, V \rangle$ où M est un *moment* et V une valeur d'un attribut au moment M. Un moment désigne soit un instant, soit un intervalle d'instant, soit encore

une séquence d'intervalles discontinus. Un historique est défini sur un domaine temporel qui peut être décomposé en deux sous-ensembles : le *domaine temporel effectif* et le *domaine temporel potentiel*. Le domaine temporel effectif est constitué de tous les instants pour lesquels une saisie de données a été effectuée. Le domaine temporel potentiel est la différence du domaine temporel avec le domaine temporel effectif. TEMPOS distingue alors deux types principaux d'historiques (cf. Figure II-15) : les historiques *en extension* et les historiques *en compréhension*. Un historique en extension est appelé *chronique* et son domaine temporel est égal à son domaine effectif. Trois types de chroniques sont définis suivant la notion de moment utilisé, par exemple une *I_chronique* est une séquence de couples associant un instant à une valeur :

I_chronique : [$\langle 1, v1 \rangle, \langle 2, v1 \rangle, \langle 3, v1 \rangle, \langle 4, v2 \rangle, \langle 5, v2 \rangle, \langle 7, v3 \rangle, \langle 8, v1 \rangle, \langle 9, v1 \rangle$]
 X_chronique : [$\langle [1...3], v1 \rangle, \langle [4...5], v2 \rangle, \langle [7...7], v3 \rangle, \langle [8...9], v1 \rangle$]
 D_chronique : [$\langle [[1...3], [8...9]], v1 \rangle, \langle [[4...5]], v2 \rangle, \langle [[7...7]], v3 \rangle$]

Un historique en compréhension associe une chronique et une fonction d'interpolation. La fonction d'interpolation est alors utilisée pour calculer la valeur de l'historique sur le domaine temporel potentiel à partir des instantanés du domaine effectif. Les historiques en escalier et discret sont traités comme des cas particuliers.

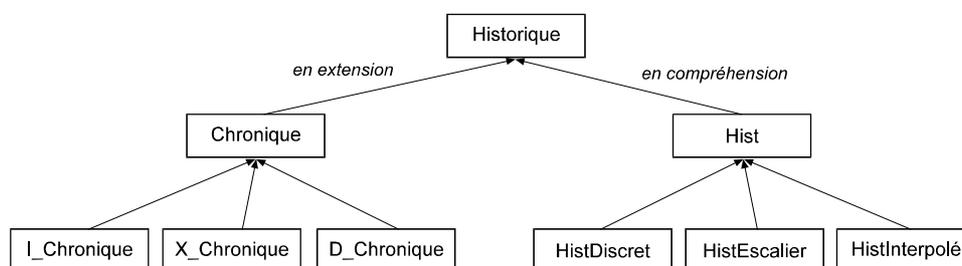


Figure II-15 : hiérarchie de types d'historiques dans TEMPOS (Fauvet *et al.*, 1997)

Le regroupement d'un ensemble d'historiques peut alors être effectué par la définition d'une classe d'objets dans laquelle des attributs constants, fugitifs (seule la dernière valeur est sauvegardée) et historiques peuvent se côtoyer :

```

    Classe Employé <
        nom : Constant (string),
        adresse : Historique (jour, string),
        salaire : Historique (mois,real),
        dateEmbauche : Fugitif (Instant(jour)),
        affectation : Historique (mois, Rayon)>
    
```

Enfin, TEMPOS définit un ensemble d'opérations sur les historiques :

- opérateurs temporels de l'algèbre relationnelle (projection temporelle, restriction temporelle et produit naturel),
- fonctions générales sur les historiques (application d'une fonction aux éléments de l'historique, sélection suivant un prédicat, découpage en deux historiques suivant un élément, construction des couples consécutifs).

1.2.3 Limites de la reconnaissance d'historiques et des bases de données temporelles

Les approches de reconnaissances d'historiques et les systèmes de gestion de bases de données temporelles n'abordent pas l'intégralité de la problématique de l'indexation telle qu'elle est nécessaire durant un cycle de raisonnement à partir de cas. Pour effectuer la comparaison des approches, nous utilisons un vocabulaire commun issu du RàPC (phases de recherche, réutilisation,

apprentissage) mais qui n'est pas utilisé à l'origine dans ces travaux. Le Tableau II-2 donne la synthèse de cette comparaison.

	<i>Raisonnement à partir de cas</i>	<i>Reconnaissance d'historiques</i>	<i>Bases de données temporelles</i>
Représentation des indices	✓	✓	✓
Mise en correspondance des indices (recherche)	✓	✓	✓ (requêtes)
Utilisation de la mise en correspondance (réutilisation)	✓	Pas d'analyse des indices	Pas d'analyse des indices
Modification de l'indexation (apprentissage)	✓	Ajout, pas de modification des indices	Ajout, pas de modification des indices

Tableau II-2 : synthèse des approches d'indexation

La reconnaissance d'historiques met l'accent sur la mise en correspondance, et les mécanismes utilisés peuvent être présentés sous la forme d'un raisonnement dont les étapes ne sont généralement pas séquentielles :

- *Recherche* : la mise en correspondance de l'historique cible avec un historique source, ou un type d'historiques sources, est effectuée et donne un résultat généralement booléen.
- *Réutilisation* : la réutilisation se borne à retourner les informations attachées à l'historique source (diagnostic par exemple) sans analyser les différences entre des historiques sources et cible.
- *Apprentissage* : l'apprentissage est uniquement effectué par l'ajout d'un nouvel historique, mais il n'y a jamais modification automatique des historiques existants.

Ces mécanismes sont insuffisants pour pouvoir gérer les indices des cas et dénotent en fait une différence d'approche plus profonde. De manière générale, les approches de reconnaissance d'historiques supposent leurs connaissances exactes et n'ont donc pas le besoin de les réviser comme dans un système de RàPC.

Les approches liées aux bases de données temporelles offrent des techniques intéressantes pour le stockage des historiques et leur manipulation par un ensemble d'opérateurs. Cependant ces systèmes n'abordent pas non plus complètement la problématique de l'indexation par historiques et le raisonnement est réduit aux mécanismes suivants :

- *Recherche* : la recherche correspond à l'exécution d'une requête qui ne retourne que les historiques répondant strictement aux contraintes exprimées,
- *Réutilisation* : la réutilisation se borne à retourner les informations attachées à l'historique source.
- *Apprentissage* : l'apprentissage est uniquement effectué par l'ajout d'un nouvel historique.

2 Proposition du « modèle d'indexation par situations comportementales »

Pour faciliter l'application de solutions récurrentes d'indexation dans le cadre de la classe de problèmes visée, nous proposons le *modèle d'indexation par situations comportementales*. Ce modèle comprend :

- le modèle de représentation proprement dit des indices et des cas,
- et un guide d'utilisation présentant certains principes de gestion des indices et des cas pour la conception d'un cycle de raisonnement.

Nous introduisons tout d'abord le modèle de représentation qui définit formellement la structuration des indices et leur intégration dans les cas. Puis, nous présentons le guide d'utilisation qui formule la synthèse des problématiques d'indexation abordées pour la nutrition des plantes et pour l'assistance à la navigation sur le Web. Ce guide a pour objectif d'une part de définir un cadre unifié qui nous servira à présenter les applications réalisées, et d'autre part de fournir une base méthodologique pour l'utilisation du modèle dans d'autres applications. Nous illustrons enfin notre modèle d'indexation avec la spécification d'un système de recherche des cas pour la nutrition des plantes. Dans cet exemple, nous vérifions également que les concepts introduits permettent de prendre en compte les caractéristiques de cette application.

2.1 Modèle de représentation des indices et des cas

Notre modèle de représentation s'appuie tout d'abord sur des hypothèses que nous précisons. Nous donnons alors une brève description du modèle pour introduire les principaux concepts. Nous décrivons enfin de manière détaillée la représentation des données d'observation, des situations comportementales et des cas.

2.1.1 Hypothèses

Les hypothèses prises par notre modèle de représentation reprennent les cinq caractéristiques déjà introduites de la classe de problèmes visée (cf. chapitre I, §3.1). Brièvement, nous rappelons que la prise en compte d'historiques est choisie ou nécessaire pour formuler les problèmes dans lesquels un processus dynamique est observé. Cette observation mène à la collecte d'un ensemble important de données brutes par échantillonnage et/ou lors d'événements. Enfin, les connaissances du domaine sont réduites et une meilleure compréhension est recherchée.

La prise en compte de ces caractéristiques se traduit au niveau des indices par deux hypothèses supplémentaires. En effet, dû au manque de connaissances du domaine et à la nécessité de prendre en compte les historiques, nous supposons que :

- Les indices effectifs d'un cas source ne peuvent pas être décrits par un ensemble de descripteurs toujours identiques. Les indices effectifs doivent principalement être extraits des historiques passés des variables d'observation, dont les valeurs sont considérées comme des indices potentiels.
- Les indices effectifs d'un cas cible ne peuvent pas être définis *a priori* : c'est au raisonnement d'analyser le cas cible pour déterminer les cas sources applicables en fonction de leurs propres indices effectifs.

2.1.2 Présentation générale

Notre modèle repose sur une représentation à deux niveaux :

- Représentation des *données brutes d'observation* sous la forme de chroniques correspondant aux évolutions des variables d'observation du processus, qui sont alors regroupées dans un enregistrement pour une période de temps définie et pour un contexte donné.
- Représentation des *connaissances* sous la forme de cas qui associent la définition d'un problème à l'expression d'une solution spécifique à la tâche de résolution de problèmes à accomplir. Le problème comprend une situation comportementale référant les données brutes des enregistrements.

Cette séparation permet une meilleure gestion des données et des indices. Premièrement, la définition des situations est modifiable par simple changement de la sélection des données brutes au sein des enregistrements. Deuxièmement, la mise en œuvre de cette séparation évite la duplication des données brutes dans les situations qui peuvent se recouvrir dans un même enregistrement.

Une situation comportementale est constituée d'une *composante instantanée* et d'une *composante comportementale*. La composante instantanée contient tous les indices instantanés liés au cas. En particulier, cette composante contient le contexte de l'enregistrement référencé. La composante comportementale constitue le centre de notre modèle d'indexation, d'où le nom de « situation comportementale ». Cette composante détermine à partir d'un instant de référence les indices effectifs d'un cas, parmi les indices potentiels correspondants aux données issues des enregistrements. Plus précisément, cette composante définit un ensemble de *comportements élémentaires* pour chaque variable, ainsi que des *contraintes temporelles* entre ces comportements. Un comportement élémentaire sélectionne une sous-séquence d'une chronique. Chaque comportement élémentaire possède une contrainte de localisation qui permet d'abstraire la raison pour laquelle ce comportement est sélectionné, par exemple : à cause des valeurs de la séquence, de sa position temporelle ou de sa position en terme du nombre d'événements par rapport à l'instant de référence. Ce formalisme de représentation permet d'exprimer des situations simples ou complexes obtenues par apprentissage ou provenant des connaissances du domaine :

- situation simple : « La situation *A* est définie à l'instant t_0 par les cinq dernières valeurs de toutes les chroniques ». Dans ce cas, les comportements élémentaires sont définis avec une contrainte de localisation par position et il n'y a pas de contraintes temporelles.
- situation complexe (Figure II-16) : « La situation *B* est définie à l'instant t_1 par les valeurs de la variable V_1 dans les 30 dernières secondes (b_1), par une forte augmentation des valeurs de la variable V_2 (b_2), et par les occurrences des événements α (b_3) et β (b_4) sur la variable V_3 ; α précède β (c_1) et β se produit pendant l'augmentation de V_2 (c_2) ». Cette situation montre également comment il est possible de prendre en compte des chroniques échantillonnées (V_2) et à événements (V_1 et V_3) grâce aux comportements élémentaires.

Une situation comportementale est alors utilisée pour déterminer les conditions de réutilisabilité d'un cas. Nous proposons dans notre modèle l'utilisation de deux types de cas permettant de représenter deux types d'enseignements :

- *cas potentiels* : enseignements basés sur des hypothèses préétablies.
- *cas concrets* : enseignements spécifiques, révisés ou atypiques.

Les cas potentiels, que nous introduisons dans nos travaux, n'ont pas d'existence explicite mais sont représentés en compréhension par un *patron*. Un patron définit des règles de construction d'un cas pour un enregistrement et un instant de référence donnés. Il définit en particulier les règles de construction de la situation comportementale. Le résultat de l'instanciation d'un patron est aussi

appelé un cas potentiel. Un cas potentiel peut être mémorisé sous la forme d'un cas concret lors de l'apprentissage.

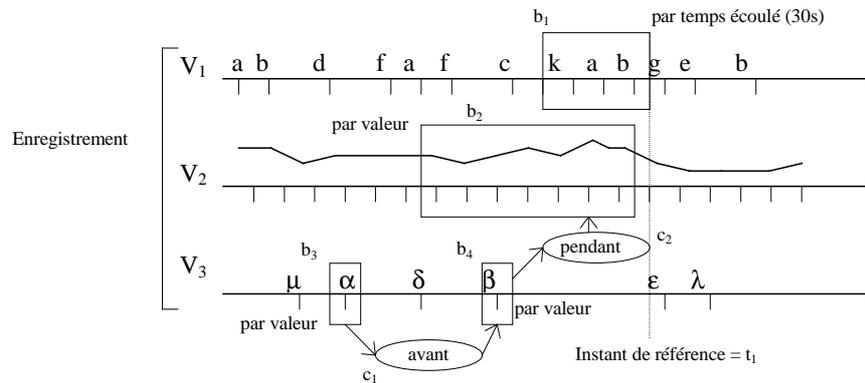


Figure II-16 : exemple de situation complexe

2.1.3 Description détaillée

Nous précisons tout d'abord la granularité des cas choisie dans notre modèle. Puis, nous présentons les deux niveaux de représentation qui séparent les données brutes d'observation stockées dans des *enregistrements*, des connaissances utiles avec deux types de cas : cas potentiels et cas concrets. Chaque cas est alors lié à une *situation comportementale* qui définit l'ensemble de ses indices effectifs par la sélection de certaines données dans un enregistrement.

2.1.3.1 Granularité des cas

Un cas est une expérience qu'il faut mémoriser et qui sera utile pour la résolution des problèmes futurs. Cependant, que représente un cas dans des problèmes complexes comme la nutrition des plantes, la navigation sur le Web, la prévision de l'évolution d'un feu de forêt ou encore la commande de robots ? Un cas représente-t-il une longue expérience ou une expérience particulière et assez précise dans le temps ? Ce problème est la généralisation du problème de la *granularité* des cas en prenant en compte la dimension temporelle : un cas est-il monolithique ou décomposable ? Un cas représente-t-il un unique enseignement ou un ensemble d'enseignements utiles ?

Par exemple, le système de raisonnement à partir de cas CELIA (Redmond, 1990) propose la décomposition d'un cas en morceaux (*snippet*). Chaque morceau est indexé séparément, et les morceaux d'un même cas sont liés entre eux. Le cas global est également indexé et contient des informations générales. Cette décomposition permet de retrouver des similarités entre des parties de cas différents et ainsi peut permettre la généralisation d'enseignements (Kolodner, 1993, pages 182–186). A l'inverse, REBECAS (Rougegrez, 1994a) propose une représentation avec un très gros grain : un cas contient tous les événements d'un feu depuis son début jusqu'à sa fin. REBECAS évite ainsi le stockage d'enseignements plus précis, mais l'inconvénient est immédiat car aucune leçon n'est tirée des différents raisonnements comme nous l'avons montré (cf. §1.1.2.2.2). Nous pensons que la prise en compte d'historiques ne fait que rendre plus nécessaire le besoin de représenter des expériences spécifiques pour améliorer notamment les possibilités d'apprentissage et l'efficacité de la recherche. C'est pourquoi nous appelons *cas* un enseignement à un instant précis, issu d'une longue expérience représentée dans notre modèle par un enregistrement. Cet enseignement est alors identifié par une situation comportementale.

Toutefois, la gestion de cas ayant une granularité temporelle fine implique la représentation de nombreux cas. Notre modèle d'indexation propose alors :

- une représentation à deux niveaux (enregistrement et cas) permettant de partager les données brutes d'observation nécessaires à l'expression des cas, pour minimiser les ressources nécessaires,
- deux types de cas (potentiels et concrets) permettant de ne pas représenter de manière explicite et exhaustive tous les cas.

2.1.3.2 Représentation des données d'observation

Les données d'observation sont représentées par des historiques de variables, appelés *chroniques*. Un *enregistrement* regroupe alors un ensemble de chroniques :

- pour une période de temps définie et
- pour un contexte donné.

Il est en effet courant de pouvoir identifier des périodes d'observation ou des sessions. L'observation du processus est alors segmentée dans le temps vis-à-vis d'un contexte donnant les caractéristiques constantes d'un enregistrement : la localisation de l'expérience, les modes opératoires ou encore les objectifs. Le contexte peut également comporter des informations de synthèse issues des chroniques (par exemple des statistiques sur les valeurs des variables).

Les données d'un enregistrement sont *brutes* par rapport à la représentation de haut niveau nécessaire pour les indices. Cependant, ces données qui sont issues de capteurs et de mesures font en général l'objet d'un prétraitement. De plus, certaines données peuvent être calculées ou déduites à partir des mesures de base. Nous n'aborderons pas ici les problèmes de traitement du signal, bien que ces techniques puissent être utilisées en amont (pour traiter le bruit et les valeurs manquantes par exemple). Ainsi, nous supposons que le processus dynamique et son environnement sont observés suivant un ensemble de variables caractéristiques de leurs états qui sont potentiellement pertinentes vis-à-vis de la tâche effectuée par le système de raisonnement. Ces variables d'observation prennent des valeurs à différents instants utilisant une unité de temps propre.

Sur le plan du formalisme de représentation des historiques, nous optons pour la réutilisation de certains éléments proposés dans le système de gestion de bases de données temporelles TEMPOS (cf.§1.2.2.2). TEMPOS offre un cadre de base que nous enrichissons des opérateurs nécessaires à la gestion des indices de notre modèle d'indexation²⁰. Cet outil se limite en effet au stockage et à des manipulations restreintes des historiques.

Dans TEMPOS, l'observation au cours du temps d'une variable est représentée par un historique en extension, appelé chronique, qui contient les données réellement saisies. Au-dessus de cette représentation, un historique en compréhension est défini : historique discret, historique en escalier ou historique interpolé. Ce type d'historiques interprète les valeurs manquantes sur le domaine temporel défini. Nous introduisons alors un troisième niveau : *l'historique en indices potentiels*. Ce type d'historiques est constitué des observations pertinentes d'une variable. Par exemple (cf. Figure II-17), un historique en indices potentiels peut ne retenir que les extremums de la fonction interpolée par l'historique en compréhension. De même, un historique en indices potentiels peut ne retenir que les événements dénotant un changement d'état (élimination des mesures successives identiques), nous obtenons alors un *historique concis* (Williams, 1986). Dans le cas le plus simple, les historiques en indices potentiels, en compréhension et en extension sont identiques.

²⁰ Nous désirions ainsi faciliter la perspective de l'intégration de cet outil comme un élément de base pour le stockage des historiques utilisés dans notre modèle d'indexation.

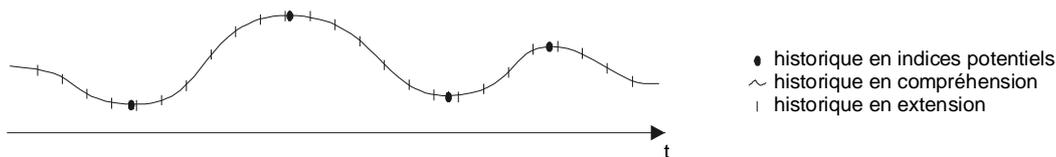


Figure II-17 : exemple d'historique en indices potentiels

Dans la suite, nous utilisons la notation suivante²¹, basée sur celle de TEMPOS : $\langle \rangle$, $[]$, et \rightarrow désignent respectivement les types des constructeurs de tuples, de séquences (suite ordonnée de valeurs) et de fonctions. $T_1(T_2)$ dénote une instance du type T_1 paramétrée par le type T_2 . L'opérateur \leftarrow désigne l'énumération d'une séquence. Dans TEMPOS, le type d'historiques le plus général d'une variable est noté $\text{Historique}(u,T)$ pour l'unité temporelle u et le type de valeur structurelle T . Nous étendons cette modélisation en ajoutant la fonction TS qui retourne l'historique en indices potentiels d'un historique, sous la forme d'une chronique d'instant :

$$\text{TS} : \text{Historique}(u, T) \rightarrow \text{I_chronique}(u,T)$$

Pour des raisons de simplification, nous désignerons dans la suite de notre document sous le terme *chronique* le résultat de la fonction TS appliquée à un historique de TEMPOS et nous renommons $\text{TSchronique}(u,T)$ son type (qui est $\text{I_chronique}(u,T)$). Nous avons alors les propriétés et les fonctions suivantes dans lesquelles $\text{Instant}(u)$ dénote le type des instants de l'unité u :

$$\text{TSchronique}(u,T) : [\langle \text{Instant}(u), T \rangle]$$

$$\{ \text{soit } C = [C_1, \dots, C_n] \text{ de type } \text{TSchronique}(u,T), \forall k \in [1..n-1], \text{VT}(C_k) < \text{VT}(C_{k+1}) \}$$

$$\text{DomT} : \text{TSchronique}(u,T) \rightarrow [\text{Instant}(u)]$$

$$\{ \text{domaine temporel} : \text{DomT}(K) = [i \mid \langle i, v \rangle \leftarrow K] \}$$

$$\text{VS} : \langle \text{Instant}(u), T \rangle \rightarrow T$$

$$\{ \text{valeur structurelle} : \text{VS}(\langle i, v \rangle) = v \}$$

$$\text{VT} : \langle \text{Instant}(u), T \rangle \rightarrow \text{Instant}(u)$$

$$\{ \text{valeur temporelle} : \text{VT}(\langle i, v \rangle) = i \}$$

Différents types de chroniques peuvent être définis suivant les propriétés de leur domaine temporel. Nous distinguons notamment les *chroniques échantillonnées* et les *chroniques à événements* (cf. Figure II-18). Si la durée séparant deux valeurs temporelles successives du domaine temporel est toujours constante, nous parlerons alors d'une chronique échantillonnée (périodique). Si la chronique n'est pas échantillonnée, nous parlerons de chronique à événements (apériodique), où un événement est une information datée et instantanée extraite du déroulement du processus. Ces événements peuvent avoir différentes sémantiques : alarmes, début ou fin d'une action, déroulement d'une action instantanée pour l'unité de temps considérée, changement d'état.

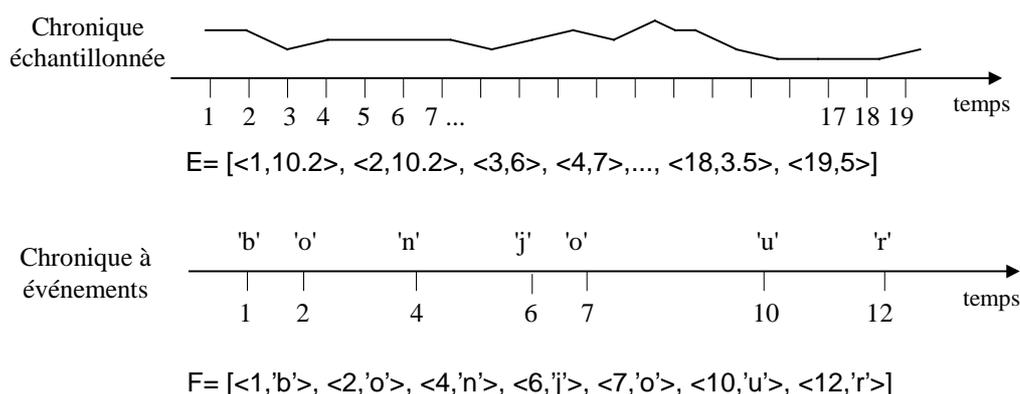
$$\text{ChroniqueEch}(u,T) : \text{TSchronique}(u,T)$$

$$\{ \text{soit } C = [C_1, \dots, C_n] \text{ de type } \text{TSchroniqueEch}(u,T), \\ \exists d \text{ tel que } \forall k \in [1..n-1], \text{VT}(C_k) - \text{VT}(C_{k+1}) = d \}$$

$$\text{ChroniqueEvt}(u,T) : \text{TSchronique}(u,T)$$

$$\{ \text{soit } C = [C_1, \dots, C_n] \text{ de type } \text{TSchroniqueEvt}(u,T), \\ \neg (\exists d \text{ tel que } \forall k \in [1..n-1], \text{VT}(C_k) - \text{VT}(C_{k+1}) = d) \}$$

²¹ Cette notation repose sur des notions de programmation fonctionnelle (Bird & Wadler, 1988 ; Scholl *et al.*, 1993).


 Figure II-18 : exemples de chroniques échantillonnées et à événements²²

Plusieurs opérateurs peuvent être définis sur les chroniques en se basant, soit sur des opérations temporelles (projection, restriction, produit naturel), soit sur des opérations sur les séquences. Nous donnons uniquement la définition des huit opérateurs de base qui sont nécessaires pour manipuler les données d'observation et définir les indices d'un cas. Seuls les deux premiers opérateurs sont présents dans TEMPOS.

- Restriction temporelle sur les instants :

$$\Gamma \quad : \text{TSChronique}(u, T), [\text{Instant}(u)] \rightarrow \text{TSChronique}(u, T)$$

$$\{ \Gamma(K, R) = [\langle i, v \rangle \mid \langle i, v \rangle \leftarrow K; i \in R] \}$$

- Découpage de la séquence suivant un prédicat (l'élément vérifiant le prédicat est exclu de la séquence retournée) :

$$\text{Début} \quad : \text{TSChronique}(u, T), (\langle \text{Instant}(u), T \rangle \rightarrow \text{Booléen}) \rightarrow \text{TSChronique}(u, T)$$

$$\{ \text{Début}([], P) = []$$

$$\text{Début}(e, S, P) = \text{si } P(e) \text{ alors } [] \text{ sinon } e.\text{Debut}(S, P) \}$$

- Longueur d'une chronique :

$$\text{Len} \quad : \text{TSChronique}(u, T) \rightarrow \text{Entier}$$

$$\{ \text{Len}([]) = 0; \text{Len}([C_1, \dots, C_n]) = n \}$$

- Identification d'un élément par sa position :

$$\text{Index} \quad : \text{TSChronique}(u, T), \text{Entier} \rightarrow \langle \text{instant}(u), T \rangle$$

$$\{ \text{Index}([C_1, \dots, C_k, \dots, C_n], k) = C_k \text{ si } 1 \leq k \leq n, \text{ nil sinon} \}$$

- Restriction par position :

$$\Gamma_P \quad : \text{TSChronique}(u, T), [\text{Entier}] \rightarrow \text{TSChronique}(u, T)$$

$$\{ \Gamma_P(K, R) = [e \mid j \leftarrow R; e = \text{Index}(K, j)] \}$$

- Position du dernier élément dont la valeur temporelle est inférieure ou égale à un instant :

$$\text{Pos} \quad : \text{TSChronique}(u, T), \text{Instant}(u) \rightarrow \text{Entier}$$

$$\{ \text{Pos}(K, t) = \text{Len}(\text{Début}(K, f_t)) \text{ avec } f_t(\langle i, v \rangle) = \text{vrai si } i > t \text{ sinon faux} \}$$

- Instants de début et de fin d'une chronique :

$$T_{\min} \quad : \text{TSChronique}(u, T) \rightarrow \text{Instant}(u)$$

$$\{ T_{\min}([]) = \text{nil}; T_{\min}(K) = \text{VT}(\text{Index}(K, 1)) \}$$

$$T_{\max} \quad : \text{TSChronique}(u, T) \rightarrow \text{Instant}(u)$$

$$\{ T_{\max}([]) = \text{nil}; T_{\max}(K) = \text{VT}(\text{Index}(K, \text{Len}(K))) \}$$

²² Dans ces exemples, l'ensemble des instants est considéré comme isomorphe aux entiers.

Par exemple, l'application de ces opérations à la chronique à événements F, décrite dans la Figure II-18, donne les résultats suivants :

$$\begin{aligned} \Gamma(F, [5...11]) &= \Gamma(F, [6...10]) = [<6, 'j'>, <7, 'o'>, <10, 'u'>] \\ \text{Debut}(F, g) &= [<1, 'b'>, <2, 'o'>, <4, 'n'>] \text{ avec } g(<i, v>) = \text{vrai si } v='j' \text{ faux sinon} \\ \text{Len}(F) &= 7 \\ \text{Index}(F, 3) &= <4, 'n'> \\ \text{Pos}(F, 9) &= \text{Pos}(F, 7) = 5 \\ T_{\min}(F) &= 1 \quad T_{\min}(\Gamma(F, [6...10])) = 6 \\ T_{\max}(F) &= 12 \end{aligned}$$

2.1.3.3 Représentation des situations comportementales

Un enregistrement peut regrouper un ensemble considérable de données brutes issues d'observations directes ou de traitements de ces observations pour une période de temps qui peut être assez étendue (plusieurs mois dans le cas de la nutrition des plantes). Nous supposons également que, pour une application donnée, les chroniques d'un enregistrement sont exprimées dans une même unité²³. Nous proposons un modèle de représentation de *situations comportementales* permettant la sélection des indices effectifs d'un cas en référant un enregistrement à un instant donné. Nous introduisons tout d'abord la structure générale d'une situation comportementale puis nous détaillons ses trois types d'éléments : d'une part les comportements élémentaires et leurs contraintes de localisation, et d'autre part les contraintes temporelles. Ces éléments peuvent être manipulés durant le raisonnement à travers des interfaces fonctionnelles que nous décrivons enfin.

2.1.3.3.1 Structure générale d'une situation comportementale

Le caractère prédictif d'un indice est très important pour la réutilisation d'un cas. Par exemple pour la nutrition des plantes, une décision est prise chaque jour pour déterminer la concentration de la solution de l'irrigation. Un indice prédictif est une caractéristique qui conditionne le choix de cette concentration. Un indice a donc un caractère explicatif qui peut être réutilisé dans les raisonnements. Si les indices ne caractérisent pas suffisamment un cas, celui-ci sera appliqué dans d'autres conditions et mènera à des échecs. Si les indices caractérisent trop précisément un cas, il ne pourra jamais être réutilisé. Notre modèle d'indexation n'aborde pas les méthodes permettant de déterminer les indices, il se limite à offrir un cadre de représentation et de gestion.

Pour la prise en compte d'historiques dans une situation, nous proposons la notion de *situation comportementale* permettant d'identifier un cas parmi les indices potentiels formés de toutes les données d'un enregistrement. Une situation comportementale est constituée :

- d'une *composante instantanée* qui comprend le contexte de l'enregistrement et toutes les données à caractère instantané qui peuvent être définies pour une application particulière,
- d'une *composante comportementale* qui définit un ensemble de *comportements élémentaires* pour chaque variable, ainsi que des *contraintes temporelles* entre ces comportements. De plus, chaque comportement élémentaire possède une *contrainte de localisation* qui explicite la raison pour laquelle il a été sélectionné.

La composante comportementale est basée sur la donnée d'un instant précis dans un enregistrement, appelé *instant de référence*. Cet instant détermine, pour la situation considérée, la limite entre d'une part le futur (instants strictement supérieurs) et d'autre part le passé et le présent. La composante comportementale décrit, de manière relative par rapport à cet instant, les évolutions

²³ L'étude de l'intégration de la multi-granularité du temps de TEMPOS reste en perspective de notre travail.

pertinentes des variables (comportements élémentaires et contraintes de localisation) et leurs relations (contraintes temporelles).

2.1.3.3.2 Comportements élémentaires et contraintes de localisation

Un comportement élémentaire définit une *sous-chronique* d'une chronique liée à une variable d'observation. Une sous-chronique $K(n,m)$ est une chronique constituée d'une sous-séquence des éléments compris entre les positions n et m d'une chronique K . Une sous-chronique est définie par l'application de l'opérateur de restriction par position :

$$K(n,m) = \Gamma_P(K,[n..m])$$

Associée à un comportement élémentaire, une contrainte de localisation explicite la *raison* pour laquelle ce comportement est sélectionné. Grâce aux contraintes de localisation, il est possible d'identifier, dans les données d'observation d'un nouveau problème et par rapport à l'instant courant, les comportements pertinents *homologues* à une situation comportementale donnée. Cette étape d'identification menée au cours du raisonnement permettra alors de décider si le cas source lié à cette situation peut être réutilisé. Nous avons défini quatre types de fonctions de localisation (cf. Figure II-19) :

- *par position* : la sous-chronique est identifiée par le nombre de mesures depuis l'instant de référence,
- *par temps écoulé* : la sous-chronique est identifiée par le temps écoulé relativement à l'instant de référence,
- *par valeur* : la sous-chronique est identifiée par sa succession de valeurs,
- *par référence* : la sous-chronique est identifiée par référence à une sous-chronique existante.

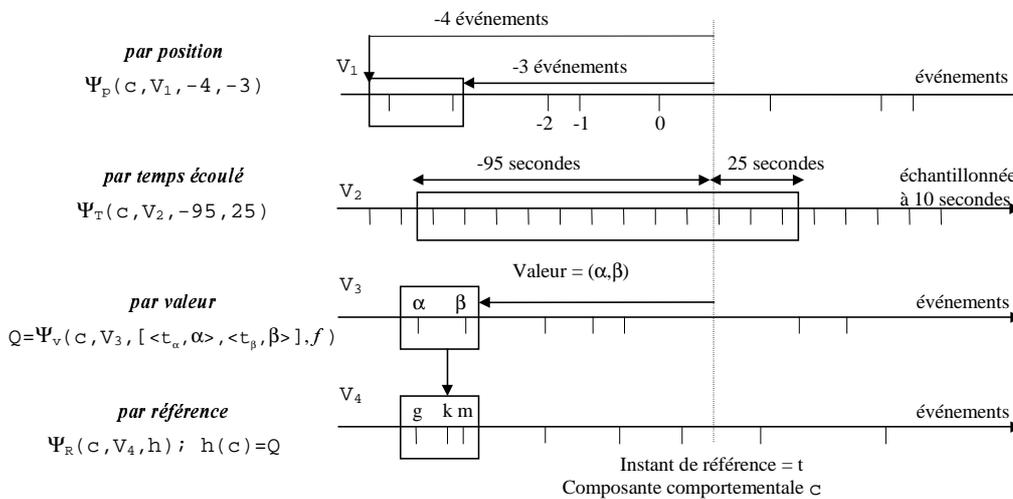


Figure II-19 : exemples de contraintes de localisation

Ces fonctions de localisation peuvent être formalisées en utilisant les opérateurs de base sur les chroniques. Nous définissons tout d'abord deux fonctions permettant de récupérer la chronique d'une composante comportementale pour une variable donnée (Chr) et l'instant de référence (Ref) :

Chr : ComposanteComportementale, Variable \rightarrow TChronique(u,T)

Ref : ComposanteComportementale \rightarrow Instant(u)

Chaque fonction de localisation prend comme paramètre la composante comportementale considérée, notée c , et la variable sur laquelle porte la localisation, notée v . Les fonctions utilisent alors d'autres paramètres suivant l'opération à effectuer. La fonction Ψ_P permet d'identifier une sous-

chronique par position, étant donné les nombres de mesures dp_1 et dp_2 relativement à l'instant de référence.

$$\Psi_P : \text{ComposanteComportementale, Variable, Entier, Entier} \rightarrow \text{TSchronique}(u,T)$$

$$\{ \Psi_P(c,v, dp_1, dp_2) = \Gamma_P(\text{Chr}(c,v), [\text{Pos}(\text{Chr}(c,v), \text{Ref}(c)) + dp_1 \dots \text{Pos}(\text{Chr}(c,v), \text{Ref}(c)) + dp_2]) \}$$

La fonction Ψ_T permet d'identifier une sous-chronique par temps écoulé relativement à l'instant de référence, étant donné les durées dt_1 et dt_2 .

$$\Psi_T : \text{ComposanteComportementale, Variable, Durée}(u), \text{Durée}(u) \rightarrow \text{TSchronique}(u,T)$$

$$\{ \Psi_T(c,v, dt_1, dt_2) = \Gamma(\text{Chr}(c,v), [\text{Ref}(c) + dt_1 \dots \text{Ref}(c) + dt_2]) \}$$

La fonction Ψ_V permet d'identifier une sous-chronique par valeur, étant donné une chronique de référence Q et une fonction de reconnaissance f . Cette fonction de localisation permet de parcourir en arrière une chronique jusqu'à ce que la fonction f reconnaisse la sous-séquence.

$$\Psi_V : \text{ComposanteComportementale, Variable, TSchronique}(u,T),$$

$$(\text{TSchronique}(u,T), \text{TSchronique}(u,T) \rightarrow \text{Booléen}) \rightarrow \text{TSchronique}(u,T)$$

$$\{ \Psi_V(c,v, [], f) = []; \Psi_V(c,v, Q, f) = g(\Gamma_P(\text{Chr}(c,v), 1, \text{Pos}(\text{Chr}(c,v), \text{ref}(c))), Q, f)$$

$$g(W, Q, f) = [] \text{ avec } \text{Len}(W) < \text{Len}(Q)$$

$$g(S.W.e, Q, f) = W.e \text{ si } f(W.e, Q) \text{ sinon } g(S.W, Q, f) \text{ avec } \text{Len}(W.e) = \text{Len}(Q) \}$$

La fonction Ψ_R permet d'identifier une sous-chronique d'une chronique K , étant donné une fonction h donnant la chronique déjà identifiée dans la composante comportementale.

$$\Psi_R : \text{ComposanteComportementale, Variable,}$$

$$(\text{ComposanteComportementale} \rightarrow \text{TSchronique}(u,T)) \rightarrow \text{TSchronique}(u,T)$$

$$\{ \Psi_R(c,v, h) = \Gamma(\text{Chr}(c,v), [\text{T}_{\min}(h(c)) \dots \text{T}_{\max}(h(c))]) \}$$

Chaque comportement élémentaire, noté b_i , est alors défini par une chronique d'origine (ts), par sa localisation dans la situation d'origine (loc), et par sa contrainte de localisation (lc) à utiliser pour la mise en correspondance. Une contrainte de localisation est représentée par une fonction de localisation dont la composante comportementale est inconnue (notée '.'). Ce paramètre sera déterminé lors de l'utilisation de la contrainte. La localisation courante (loc) représente l'utilisation de la contrainte de localisation dans la composante comportementale considérée.

En reprenant l'exemple de situation comportementale de la Figure II-16, nous pouvons exprimer les comportements élémentaires et leurs contraintes de localisation sous la forme suivante, étant donné les chroniques K_1 , K_2 et K_3 correspondant respectivement aux variables V_1 , V_2 et V_3 :

$$\text{ComportementsElémentaires} = \{$$

$$b_1 = \{ts = K_1, loc = K_1(8, 10), lc = \Psi_T(., V_1, -30, 0)\},$$

$$b_2 = \{ts = K_2, loc = K_2(7, 13), lc = \Psi_V(., V_2, loc, f)\},$$

$$b_3 = \{ts = K_3, loc = K_3(2, 2), lc = \Psi_V(., V_3, [t_{\alpha}, \alpha], f)\},$$

$$b_4 = \{ts = K_3, loc = K_3(4, 4), lc = \Psi_V(., V_3, [t_{\beta}, \beta], f)\}$$

Nous avons ainsi proposé quatre types de contraintes de localisation mais cette liste n'est pas exhaustive, et le formalisme utilisé permet d'exprimer d'autres types de contraintes adaptés à la gestion d'indices dans une application spécifique.

2.1.3.3.3 Contraintes temporelles

La définition des comportements pertinents élémentaires permet d'obtenir un ensemble d'éléments de même nature, bien que les chroniques de l'enregistrement puissent être différentes (échantillonnées ou à événements). Ces comportements peuvent alors être reliés par des contraintes temporelles entre

intervalles (relations d'Allen (Allen, 1984) cf. Figure II-20). Chaque comportement élémentaire $K_{n,m}$ définit en effet un intervalle temporel $[T_{\min}(K(n,m))...T_{\max}(K(n,m))]$.

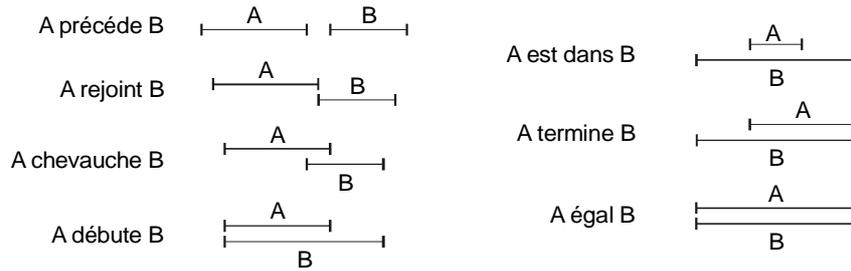


Figure II-20 : relations entre intervalles

Si les comportements élémentaires utilisés sont réduits à un élément, ils représentent des instants. Dans ce cas, les contraintes entre instants doivent être utilisées. Les intervalles représentant les comportements élémentaires sont alors décomposés en un instant de début et un instant de fin. Les contraintes temporelles peuvent porter sur des comportements élémentaires issus de variables différentes. En reprenant l'exemple de la Figure II-16, nous pouvons exprimer les contraintes temporelles pour former la description complète de la composante comportementale :

```
ComposanteComportementale= {
  ComportementsElémentaires = {
    b1={ts=K1, loc=K1(8,10), lc=Ψ_T(.,V1,-30,0)},
    b2={ts=K2, loc=K2(7,13), lc=Ψ_V(.,V2,loc,f)},
    b3={ts=K3, loc=K3(2,2), lc=Ψ_V(.,V3,[<t_α>,f]),
    b4={ts=K3, loc=K3(4,4), lc=Ψ_V(.,V3,[<t_β>,f])},
  ContraintesTemporelles= { c1=avant(b3,b4), c2=pendant(b4,b2)}
```

2.1.3.3.4 Interfaces fonctionnelles

Nous distinguons quatre interfaces fonctionnelles permettant de définir les manipulations possibles des indices comportementaux d'un cas durant le raisonnement. Pour la phase de recherche, *l'interface de consultation* est utilisée pour accéder aux valeurs des indices des cas sources pour pouvoir les comparer suivant des mécanismes spécifiques aux méthodes de recherche utilisées. Lors de la réutilisation, *l'interface d'analyse* est utilisée pour obtenir les différences entre deux composantes comportementales. *L'interface de sélection* permet principalement à la méthode d'apprentissage de pouvoir sélectionner, parmi les indices potentiels, les indices effectifs du cas cible à mémoriser. Si l'organisation de la mémoire doit être mise à jour, *l'interface de modification* permet d'ajuster l'indexation de chaque cas. Les interfaces permettent de manipuler les *éléments* de la composante comportementale : comportements élémentaires, contraintes de localisation et contraintes temporelles.

- L'interface de consultation permet l'accès aux comportements élémentaires et aux contraintes temporelles d'une composante comportementale. De plus, cette interface permet l'accès à la définition d'un comportement élémentaire : chronique référencée, sous-chronique sélectionnée et contrainte de localisation
- L'interface d'analyse permet de lister les comportements élémentaires et les contraintes temporelles communs entre deux composantes comportementales. Cette interface permet également de connaître les comportements élémentaires et les contraintes temporelles présents dans une première composante comportementale et absents dans une seconde.
- L'interface de sélection permet l'ajout de comportements élémentaires et de contraintes temporelles dans une composante comportementale.
- L'interface de modification permet la suppression des comportements élémentaires et des contraintes temporelles ainsi que la modification des contraintes de localisation : par

exemple extension ou diminution de la sous-séquence sélectionnée. La suppression d'un comportement élémentaire entraîne la suppression des contraintes temporelles associées.

2.1.3.4 Représentation des cas

Nous présentons la représentation d'un cas cible qui intègre le problème courant, puis nous détaillons les deux types de cas sources définis.

2.1.3.4.1 Cas cible

Nous rappelons qu'un cas cible est le cas en cours d'étude sur lequel le raisonnement porte. Il est constitué du problème courant à résoudre et d'une solution inconnue que le raisonnement doit déterminer. Un cas cible référence un instant précis, appelé *instant courant*, à l'intérieur de l'enregistrement courant. La composante comportementale du cas cible est vide : les indices comportementaux effectifs du cas cible sont inconnus. Toutes les données disponibles (observations et estimations) sont considérées comme des indices potentiels. Le raisonnement a donc pour but de chercher des cas sources ayant des indices effectifs pouvant être mis en correspondance parmi les indices potentiels du cas cible.

2.1.3.4.2 Deux types de cas sources

Nous proposons l'utilisation de deux types de cas : *cas potentiels* et *cas concrets*. Un cas potentiel est une notion que nous introduisons pour prendre en compte les situations non explicites, c'est-à-dire que la situation d'un cas potentiel devra être explicitée lors d'un raisonnement. Un cas potentiel n'a pas d'existence propre, mais il est défini à travers un *patron de cas potentiels*. Un patron est considéré comme un ensemble de cas potentiels en compréhension. Les patrons sont utilisés lors de la phase de recherche pour expliciter des connaissances jusqu'alors cachées dans l'ensemble des données des enregistrements. Un patron peut être en effet instancié pour un instant de référence et, par abus, nous appelons également « cas potentiel » le résultat de cette instanciation qui reste éphémère. Si le résultat de cette instanciation est ajouté à la mémoire du système lors de l'apprentissage, nous parlerons alors de cas concrets (cf. §2.2.3). Un patron P définit un prédicat (applicable_P) indiquant s'il est applicable à un instant donné dans un enregistrement et une fonction d'instanciation (nouveau_P) :

$\text{applicable}_P : \text{Enregistrement}, \text{Instant}(u) \rightarrow \text{Booléen}$
 $\text{nouveau}_P : \text{Enregistrement}, \text{Instant}(u) \rightarrow \text{CasPotentiel}$

Les cas concrets sont identifiés par des situations explicites et dynamiques. Ainsi, la représentation d'un cas concret peut être modifiée (notamment avec l'interface de modification des indices) au cours des échecs et des succès des raisonnements. Ces modifications permettront de retrouver les cas concrets pour des raisonnements dans lesquels ils seront plus utiles. Un cas concret porte des informations quant à son origine : définition manuelle ou patron utilisé pour son instanciation.

2.2 Guide d'utilisation de notre modèle de représentation

Pour l'utilisation de notre modèle de représentation des indices et des cas, nous avons identifié quatre étapes importantes :

1. Identification des variables d'observation.
2. Détermination de la sémantique d'un enregistrement et de son contexte.
3. Définition de la représentation des cas et des situations comportementales.
4. Conception des phases de raisonnement.

Pour la dernière étape qui est fortement liée au processus de raisonnement à partir de cas, nous présentons des principes de gestion des indices permettant :

- ❑ de définir un *cadre unifié* pour la description des phases de raisonnement des applications réalisées afin de simplifier leur compréhension (recherche des cas pour la nutrition des plantes cf. §2.3, et assistance à la navigation cf. chapitre V, §2),
- ❑ de fournir une *base méthodologique* facilitant l'utilisation du modèle dans la conception des phases de raisonnement pour d'autres applications.

Nous proposons ainsi des principes de gestion portant sur les phases de recherche, de réutilisation et d'apprentissage. Ces principes présentent des typologies de traitements et de données propres à notre modèle de représentation. Ils pourront être mis en œuvre suivant les connaissances et les caractéristiques d'un domaine d'application considéré.

2.2.1 Principes de recherche

Les principes de recherche reposent premièrement sur cinq types de filtrage. Ces étapes doivent alors être assemblées suivant les contraintes spécifiques d'une application. Notre modèle d'indexation laisse ainsi ouvert le contrôle de la recherche, tout en définissant des étapes de base. De plus, nous proposons une *stratégie séquentielle de recherche* qui donne un exemple typique d'assemblage de ces étapes.

2.2.1.1 Types de filtrage pour la recherche

Nous rappelons qu'un filtrage est une étape de la phase de recherche : étant donné un ensemble de cas sources et un cas cible, l'exécution d'un filtrage retourne un sous-ensemble des cas sources jugés les plus utiles pour le raisonnement au stade considéré de la recherche. Au niveau connaissance, nous avons défini cinq types principaux de filtrage :

- ❑ *Filtrage sur la composante instantanée.* Tout comme dans PADIM (Fuchs *et al.*, 1995) et RADIX (Corvaisier *et al.*, 1998), un filtrage suivant la composante instantanée peut être effectué. La composante instantanée comprend le contexte de l'enregistrement référencé, et il est souvent utile d'éliminer de la recherche les cas ayant un contexte complètement différent, car les connaissances d'adaptation peuvent être inexistantes ou difficiles à mettre en œuvre.
- ❑ *Filtrage sur les comportements élémentaires.* L'ensemble des comportements élémentaires est pris en compte pour la mise en correspondance des cas sources avec le cas cible. Avant chaque comparaison, une étape d'identification des comportements élémentaires doit être effectuée. En effet, à partir des contraintes de localisation de chaque comportement, les comportements homologues dans le cas cible sont identifiés (par position, par temps écoulé, par valeur ou par référence). Ensuite la mise en correspondance entre les comportements élémentaires eux-mêmes est effectuée puis agrégée. La sélection et le tri des cas sources peuvent prendre en compte le nombre de comportements élémentaires constituant chaque situation comme un critère supplémentaire. En effet, il peut être nécessaire de privilégier la sélection d'un cas formé d'une situation complexe (donc plus spécifique) à un cas formé de peu de comportements élémentaires.
- ❑ *Filtrage sur une restriction.* Une restriction rassemble les notions de point de vue de REBECAS (Rougegrez, 1994a) (restriction sur les variables) et de tendance de ICONS (Schmidt *et al.*, 1996) (restriction sur des horizons temporels). Une restriction est un ensemble de contraintes de localisation permettant d'identifier des comportements élémentaires jugés pertinents dans tous les cas. Une restriction permet donc d'éviter la définition de ces comportements dans chaque situation comportementale. De plus, la restriction considérée est modifiable directement sans nécessiter le changement exhaustif des comportements élémentaires des cas sources. Ce filtrage permet de ne retenir que les

cas ayant un minimum de similarité sur certaines variables et sur un horizon temporel restreint.

- *Filtrage basé sur les contraintes temporelles.* Suite à l'identification des comportements élémentaires dans le cas cible pour chaque cas source pris en compte, les contraintes temporelles sont instanciées dans le contexte du problème courant. Ce filtrage doit retourner les cas sources qui respectent le mieux les contraintes temporelles (suivant le nombre de contraintes satisfaites par exemple).
- *Filtrage des patrons.* Uniquement pour la recherche des cas potentiels, cette étape permet d'obtenir les patrons qui peuvent être appliqués à l'instant courant dans l'enregistrement du cas cible (utilisation du prédicat applicable_P).

Lors de la réalisation d'un système de RàPC utilisant notre modèle d'indexation, ces étapes pourront être assemblées pour former une séquence, ou regroupées dans une même opération. Enfin, ces étapes sont alors réalisées grâce à des index, en tirant profit des propriétés des indices manipulés (domaines de valeurs, structuration). Différents index sont en effet utilisables, comme par exemple (cf. chapitre III) : index par plus proches voisins (linéaire ou optimisé), index par hiérarchie discriminante, index par table de hachage. La plate-forme à objet CBR*Tools que nous proposons (cf. chapitres III et IV) permet de réaliser et d'intégrer ces index dans un système de RàPC.

2.2.1.2 Exemple d'une stratégie séquentielle de recherche

La stratégie séquentielle de recherche que nous proposons est basée sur les deux heuristiques suivantes :

- il est préférable de réutiliser des cas concrets plutôt que des cas potentiels, car les cas concrets peuvent avoir été révisés, reliés à des cas d'échec et/ou de succès, ou porter des informations spécifiques quant à leur réutilisation ;
- il n'est pas nécessaire de chercher les cas sources les plus utiles, mais il suffit d'identifier les cas satisfaisant un critère minimum d'utilité (seuil de similarité par exemple).

Suivant ces deux heuristiques, notre stratégie se décompose en deux phases séquentielles : recherche des cas concrets et recherche des cas potentiels. Le résultat global de la recherche correspond à celui de la première phase dont le résultat est non vide ou plus généralement dépassant un seuil de qualité (similarité minimale par exemple). Cette recherche séquentielle amène une meilleure efficacité puisque la recherche des cas potentiels est effectuée *au besoin*, lorsque les cas concrets sont de qualité suffisante (au sens du filtrage) pour être retenus. La recherche des cas potentiels est en effet la plus coûteuse des phases de recherche puisqu'elle nécessite l'instanciation des patrons sur les enregistrements supposés très nombreux et volumineux.

Nous proposons de combiner les types de filtrage identifiés de la façon suivante pour la recherche des cas concrets :

1. *Filtrage sur la composante instantanée.* La composante instantanée contient généralement des données représentant un niveau de description moins détaillé que la composante comportementale. Il s'agit donc d'effectuer en premier lieu un filtrage sur cette composante. De plus, sa structure peut être plus simple, menant à un filtrage peu coûteux et d'une incidence importante sur le nombre de cas à considérer par la suite.
2. *Filtrage sur une restriction.* Une restriction définit un ensemble d'indices pertinents dans tous les cas. Nous utilisons cette notion pour ne retenir que les cas ayant une correspondance minimum avec le cas cible. Cette étape nécessite une mise en correspondance moins complexe que le filtrage sur l'ensemble des comportements élémentaires (étape suivante).
3. *Filtrage sur les comportements élémentaires.* L'identification des comportements élémentaires et leur comparaison constituent une étape coûteuse dans le cas de situations

complexes. Toutefois, le nombre de cas sources à considérer est contrôlé par les deux étapes précédentes.

4. *Filtrage sur les contraintes temporelles.* Les contraintes temporelles permettent de comparer un cas cible et un cas source au niveau de détail le plus fin. Cette étape est donc la dernière et peut être appelée *sélection*.

Pour la recherche des cas potentiels, des étapes similaires sont appliquées avec une première étape supplémentaire permettant de filtrer les patrons à considérer :

1. *Filtrage des patrons.* Les étapes suivantes devront au besoin instancier les patrons sur les enregistrements à des instants différents afin d'effectuer la comparaison avec le cas cible. L'identification des patrons de cas potentiels applicables est donc un prérequis qui est réalisé par cette étape.
2. *Filtrage sur la composante instantanée.* A ce stade de la recherche, il est judicieux de ne prendre en compte que les indices du contexte des enregistrements. Ainsi, la comparaison des cas sources potentiels avec le cas cible ne nécessite pas l'instanciation des patrons. Cette étape réalise alors le filtrage des enregistrements.
3. *Filtrage sur une restriction.* Comme pour les cas concrets, cette étape est importante d'une part pour assurer un minimum de correspondance avec le cas cible, et d'autre part pour réduire le nombre de cas à examiner avec les étapes suivantes. De plus, cette étape ne nécessite pas l'instanciation des patrons, qui est ainsi retardée. A la fin de cette étape, les cas potentiels identifiés sont complètement instanciés (fonction `nouveauP`) et peuvent être manipulés par la suite comme des cas concrets.
4. *Filtrage sur les comportements élémentaires pertinents.* Idem aux cas concrets.
5. *Filtrage sur les contraintes temporelles.* Idem aux cas concrets.

2.2.2 Principes de réutilisation

A la suite de la phase de recherche, le système récupère :

- *le cas cible,*
- *le ou les cas sources retenus* dont les indices sont suffisamment vérifiés dans le cas cible pour estimer que chaque cas source donne une *interprétation valide* du cas cible, étant donné les connaissances du système au moment du raisonnement,
- *les traces de recherche* donnant des indications utiles pour le raisonnement et particulièrement pour la phase de réutilisation. Ces informations peuvent principalement décrire : les degrés d'utilité de chaque cas source (similarité), les interprétations du cas cible par ces cas sources (identification des comportements élémentaires et des contraintes temporelles), les résultats intermédiaires de recherche.

Avec ces informations, la phase de réutilisation doit déterminer la solution suggérée pour le cas cible courant. Si les traces de recherche intègrent les interprétations du cas cible, cette phase peut alors se baser sur la mise en correspondance pour effectuer des opérations correctrices sur les solutions proposées par les cas sources. Ces opérations sont fortement dépendantes des connaissances du domaine. Les différences entre la situation comportementale d'un cas source et la situation interprétée du cas cible sont identifiées au travers de l'interface fonctionnelle d'analyse des indices. La gestion de conflits éventuels entre les solutions proposées par les cas sources est une tâche typique de cette phase.

2.2.3 Principes d'apprentissage

L'apprentissage est la phase du raisonnement qui doit mettre à jour l'indexation de la mémoire pour améliorer le système. Quatre types d'opérations sont possibles :

- *Ajout du cas cible.* De manière classique en RàPC, le cas cible peut être mémorisé et indexé dans la mémoire comme un cas concret en définissant ses indices effectifs grâce à l'interface de sélection. Nous prenons ainsi l'hypothèse qu'à la fin du raisonnement il est possible de déterminer ses indices suivant une sélection par défaut ou suivant la solution obtenue après révision (retour d'information après application de la solution dans un environnement réel par exemple). Cet ajout permet de ne pas exécuter un raisonnement identique (qui a pu être coûteux avec une résolution de conflit par exemple) dans un futur problème similaire. L'ajout d'un cas cible nécessite l'ajout de l'enregistrement sous-jacent.
- *Ajout des cas potentiels.* Les cas potentiels identifiés lors de la phase de recherche doivent être étudiés par la phase d'apprentissage pour déterminer s'ils peuvent être utiles dans les raisonnements futurs. Dans ce cas, les cas potentiels sont mémorisés sous la forme de cas concrets. Leurs indices sont alors pris en compte dans les structures d'indexation pour des accès plus rapides et pourront être mis à jour dans les raisonnements suivants.
- *Ajout de l'enregistrement courant.* Lorsqu'un enregistrement est fermé (changement de contexte), il peut être ajouté à la mémoire si cette opération n'a pas déjà été effectuée lors de l'ajout d'un cas cible. Grâce à l'instanciation des patrons de cas potentiels effectuée au besoin avec la stratégie de recherche séquentielle, ce type d'apprentissage est dit *paresseux* . Toutefois, l'enregistrement, pour lequel les patrons ne sont pas applicables ou donneront des situations comportementales vides, ne doit pas être retenu.
- *Autre réorganisation de la mémoire et mise jour des cas.* De multiples réorganisations peuvent être déclenchées à la suite d'un raisonnement : création, modification ou élimination des cas concrets. L'objectif de cette réorganisation est d'améliorer la description des cas et de maintenir l'efficacité de la recherche.

2.3 Un exemple : recherche des cas pour la nutrition des plantes

La problématique générale de cette application a été présentée dans le premier chapitre (cf. chapitre I, §1.1). Notre objectif est d'étudier la conception d'un atelier permettant d'assister un utilisateur dans la découverte et l'analyse des connaissances liées au processus de régulation de la nutrition des plantes. Il s'agit donc d'un système permettant de manière interactive :

- de proposer des concentrations de la solution d'irrigation pour réguler le processus tout en expliquant le raisonnement effectué en se basant sur les cas utilisés,
- de modifier les situations des cas dans la phase d'apprentissage au regard des différences entre l'évolution attendue et effective du processus.

Dans cette section, nous nous plaçons dans le cadre formé par le guide d'utilisation pour présenter la conception d'un système de recherche des cas similaires pour la nutrition des plantes. A travers cette application, nous illustrons ainsi notre modèle d'indexation et nous montrons comment les cas du processus de nutrition peuvent être représentés. Cette spécification a donné lieu à la réalisation d'un prototype pour valider les représentations choisies et les mécanismes de recherche de cas similaires. L'utilisation de notre modèle d'indexation pour la réalisation d'un système complet de raisonnement est donnée au chapitre V avec notre prototype d'assistance à la navigation sur le Web.

2.3.1 Identification des variables d'observation

Dans le problème de la régulation de la nutrition des plantes, le processus à observer comprend un ensemble de plantes, l'environnement et le système de régulation. Le processus est observé par différentes variables donnant pour chaque jour (cf. Figure II-21) : la température, le rayonnement solaire, l'humidité, la concentration de la solution d'irrigation ainsi que l'écart de la concentration du drainage avec la consigne à maintenir. Les évolutions de ces variables sont représentées par des

chroniques échantillonnées numériques où l'unité de temps est le jour. Ces variables ont été choisies par les experts du domaine (après des analyses statistiques) pour leur pertinence potentielle vis-à-vis de l'objectif de la régulation qui est de maintenir stable la concentration du drainage. Le système de raisonnement à partir de cas doit alors permettre de déterminer la concentration de la solution de l'irrigation à appliquer pour le jour suivant, étant donné toutes les observations disponibles à l'instant courant.

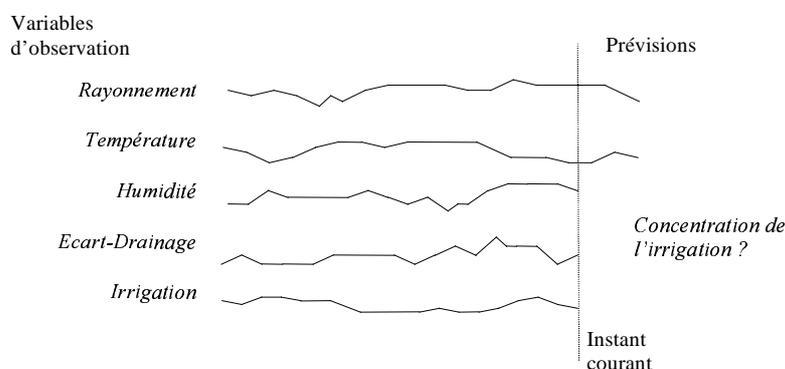


Figure II-21 : données d'observation pour la nutrition des plantes

2.3.2 Détermination de la sémantique d'un enregistrement et de son contexte

Plusieurs expérimentations de régulation sont menées et peuvent même se dérouler simultanément. Chaque expérimentation a pour objectif d'évaluer le choix pris pour la consigne de concentration du drainage à maintenir. Cette consigne est alors constante pour toute la durée de l'expérimentation. C'est pourquoi, nous regroupons les données d'observations dans des enregistrements qui représentent les données collectées pour une unique expérimentation. La consigne à maintenir définit alors le contexte de l'enregistrement (cf. Figure II-22).

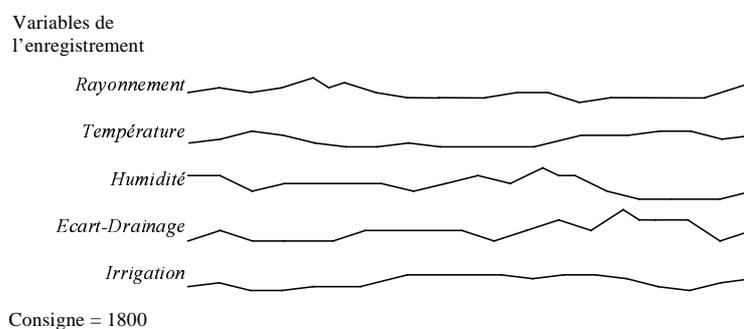


Figure II-22 : exemple d'enregistrement pour la nutrition des plantes

2.3.3 Définition de la représentation des cas

Les cas (sources et cible) sont structurés en deux composantes :

- une composante *problème* qui comprend les données d'observation et la situation comportementale (indices effectifs du cas),
- une composante *solution* qui donne la concentration de la solution d'irrigation correspondante et l'évaluation de la régulation. Cette évaluation est calculée à partir de l'évolution de la concentration de la solution de drainage dans les cinq jours suivant l'instant courant.

2.3.3.1 Cas cible

Un cas cible est constitué d'une situation comportementale vide (les indices effectifs sont indéterminés) et d'une solution que le raisonnement doit calculer :

```

CasCible {
  IdentificateurCas= "exemple-cas-cible",
  Origine = cas-cible,
  Problème= {
    Enregistrement= R, Instant= t,
    SituationComportementale= {
      ComposanteInstantanée= {ContexteEnregistrement= {Consigne= 1800}},
      ComposanteComportementale= {
        ComportementsElémentaires = {}
        ContraintesTemporelles= {}},
    Solution= {
      Commande= ?,
      Evaluation= ?}}

```

2.3.3.2 Cas potentiels

Nous définissons également deux patrons de cas potentiels : P_1 qui est applicable lorsque la moyenne du rayonnement solaire des 5 derniers jours dépasse un seuil fixé, et P_2 qui est applicable dans le cas contraire. Nous utilisons les expérimentations existantes qui montrent que pour un fort rayonnement solaire l'inertie est de 2 jours, et respectivement de 5 jours pour un rayonnement faible. C'est pourquoi le patron P_1 définit les règles de sélection des évolutions passées sur les 2 derniers jours et respectivement P_2 sur les 5 derniers jours. La fonction d'instanciation du patron P_2 s'exprime alors de la façon suivante :

```

nouveauP. (R : Enregistrement, t : Instant(u)) {
  retourne CasPotentiel {
    IdentificateurCas= générerIdentificateur(),
    Origine= P2
    Problème= {
      Enregistrement= R, Instant= t,
      SituationComportementale= {
        ComposanteInstantanée= {ContexteEnregistrement= {Consigne= Consigne(R)}},
        c=ComposanteComportementale= {
          ComportementsElémentaires = {
            b1={ts=K1, loc=ΨP(c,Rayonnement,-4,5), lc=ΨP(.,Rayonnement,-4,5)},
            b2={ts=K2, loc=ΨP(c,Température,-4,5), lc=ΨP(.,Température,-4,5)},
            b3={ts=K3, loc=ΨP(c,Humidité,-4,0), lc=ΨP(.,Humidité,-4,0)},
            b4={ts=K4, loc=ΨP(c,Ecart-Drainage,-4,0), lc=ΨP(.,Ecart-Drainage,-4,0)},
            b5={ts=K5, loc=ΨP(c,Irrigation,-4,0), lc=ΨP(.,Irrigation,-4,0)},
          ContraintesTemporelles= {}},
        Solution= {
          Commande= ΨP(c,Irrigation,1,1),
          Evaluation= ΨP(c,Ect-Drainage,1,5)}}
    }
}

```

Dans cet exemple, l'expression de la fonction d'instanciation utilise les fonctions de localisation définies. Toutefois, les règles d'instanciation d'un patron peuvent être plus complexes, que ce soit sur la partie problème ou sur la partie solution (cf. règles de sélection pour Broadway, chapitre V, §2.1.3.2). Un algorithme d'instanciation est alors nécessaire, mais il est pratique, dans un cas simple comme celui-ci, de pouvoir représenter un patron dans le même formalisme qu'un cas potentiel instancié.

2.3.3.3 Cas concrets

Un cas concret peut être issu de l'instanciation d'un cas potentiel et, grâce à sa représentation explicite, il peut être manipulé et révisé suivant les raisonnements successifs d'échec ou de succès. Il peut également être issu d'une création manuelle par les experts du domaine qui annotent ainsi une

expérience concrète de régulation. Nous donnons comme exemple un cas créé avec le patron précédent et ayant été modifié par l'utilisateur avec l'ajout d'un comportement pertinent (b_6) et d'une contrainte temporelle (c_1) (cf. Figure II-23).

```

CasConcret {
  IdentificateurCas = "exemple-cas-concret",
  Origine = P2
  Problème = {
    Enregistrement = R, Instant = t,
    SituationComportementale = {
      ComposanteInstantanée = {ContextEnregistrement = {Consigne = 1800}},
      ComposanteComportementale = {
        ComportementsElémentaires = {
          b1 = {ts = K1, loc = K1(46,55), lc = ΨP(., Rayonnement, -4,5)},
          b2 = {ts = K2, loc = K2(46,55), lc = ΨP(., Température, -4,5)},
          b3 = {ts = K3, loc = K3(46,50), lc = ΨP(., Humidité, -4,0)},
          b4 = {ts = K4, loc = K4(46,50), lc = ΨP(., Ecart-Drainage, -4,0)},
          b5 = {ts = K5, loc = K5(46,50), lc = ΨP(., Irrigation, -4,0)},
          b6 = {ts = K1, loc = K1(16,26), lc = ΨV(., Rayonnement, loc, f)},
        }
      }
      ContraintesTemporelles = { c1 = avant(b6, b1) },
    }
  }
  Solution = {
    Commande = 1200,
    Evaluation = K4(51,56)}
  }
  
```

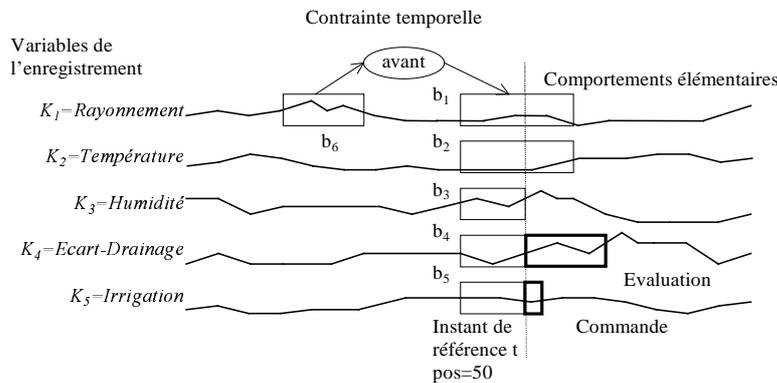


Figure II-23 : exemple de cas concret pour la nutrition des plantes

2.3.4 Conception des phases de raisonnement

Dans cet exemple, nous nous limitons à l'étude de la phase de recherche. Pour cette phase, nous utilisons notre stratégie séquentielle de recherche. La recherche des cas concrets utilise les quatre étapes que nous avons définies :

1. *Filtrage sur la composante instantanée* : la recherche est restreinte aux cas provenant d'enregistrements ayant la même consigne pour la concentration du drainage. En effet, nous n'avons pas encore de connaissances pour réutiliser les cas issus de contraintes de régulation différentes. Un index par table de hachage, permettant de ne retenir que les cas correspondant à la même consigne, est utilisé.
2. *Filtrage sur une restriction* : la restriction correspond à la sélection des évolutions passées sur les 2 derniers jours, sachant que ces données sont pertinentes dans tous les cas. Ce filtrage peut être effectué avec un index par plus proches voisins optimisé (*Kd-tree*, Wess *et al.*, 1994) utilisant une similarité numérique basée sur les écarts de valeurs. Cet index est applicable car les valeurs sont toutes numériques et leur nombre est figé.
3. *Filtrage sur les comportements élémentaires* : chaque comportement est mis en correspondance avec son comportement homologue grâce à une mesure de similarité basée sur la moyenne des écarts de chaque mesure de la chronique. Ensuite, une similarité

globale est calculée en faisant la moyenne des similarités de chaque comportement. Cette similarité est combinée avec le ratio entre le nombre de comportements d'une situation et le nombre maximum de comportements. Un index de type plus proches voisins avec un seuil de similarité minimal permet d'identifier un sous-ensemble de cas sources dont la similarité est la plus grande tout en présentant le plus de comportements élémentaires.

4. *Sélection sur les contraintes temporelles* : un index de type plus proches voisins permet d'identifier les meilleurs cas en utilisant une mesure de similarité calculant le ratio de contraintes satisfaites.

Finalement, si aucun cas concret n'est retrouvé, la recherche des cas potentiels est effectuée. Le filtrage des patrons de situations comportementales est basé sur le rayonnement solaire courant suivant les prédicats d'application des patrons. Par conception des prédicats d'applicabilité, un seul patron est sélectionné pour le raisonnement courant. Puis, les mêmes étapes que précédemment sont utilisées.

3 Conclusion

Dans ce chapitre, nous avons étudié et proposé des solutions pour la gestion des indices basés sur des historiques. Nos apports se situent à deux niveaux :

- Proposition d'une approche globale et générique de l'indexation à travers le *modèle d'indexation par situations comportementales* comprenant le modèle de représentation proprement dit des indices et des cas ainsi qu'un guide d'utilisation donnant un cadre de description et de conception des phases de raisonnement. Notre approche d'indexation pour le RàPC est la seule abordant de manière générale la représentation et la gestion d'historiques.
- Proposition d'un modèle de représentation des indices dépassant les limites des approches existantes de RàPC : représentation d'historiques hybrides, meilleure représentation pour la découverte de connaissances.

Premièrement, notre démarche est en effet originale par rapport aux travaux existants en RàPC qui restent liés à des applications particulières. Le modèle de représentation des indices et des cas est en effet générique : données d'observation regroupées dans des enregistrements et situation comportementale formée de comportements élémentaires et de contraintes temporelles. De plus, le guide d'utilisation associé à notre modèle de représentation est chargé de réunir les connaissances et les méthodes pour l'intégration effective des représentations dans les phases de raisonnement. Nous proposons un ensemble de types généraux de traitements faisant la synthèse des problématiques d'indexation étudiées (nutrition des plantes et assistance à la navigation sur le Web). Ce guide est destiné à être suivi et étendu par l'analyse de nouvelles applications ou par l'étude générale de traitements particuliers (généralisation de chroniques par exemple). La problématique du modèle d'indexation est également différente de celle abordée par les approches de reconnaissance d'historiques et les systèmes de gestion de bases de données temporelles. Il s'agit en effet de proposer des représentations et des mécanismes de gestion des indices dans un cycle de raisonnement visant la réutilisation d'expériences passées : possibilité d'analyser la mise en correspondance pour effectuer la réutilisation, possibilité de mettre à jour des cas existants.

Deuxièmement, sur le plan du modèle de représentation des indices, nous permettons tout d'abord, contrairement aux approches de RàPC étudiées, l'expression d'indices sur des chroniques échantillonnées et à événements. En effet, la sélection de comportements élémentaires crée une structure de plus haut niveau qui masque ainsi l'hétérogénéité des chroniques. Puis, nous proposons

une représentation permettant une meilleure gestion des trois étapes du processus de découverte des connaissances (cf. § 1.1.2.2) :

1. *Extraction structurée de cas potentiels à partir des données brutes.* L'extraction de cas potentiels est effectuée grâce à l'instanciation des *patrons de cas potentiels* . Les cas potentiels permettent de représenter des situations non explicites comme dans la plupart des approches existantes. Grâce à l'utilisation originale de patrons, nous permettons en plus l'expression d'une *structuration* (comportements élémentaires et contraintes temporelles) par l'utilisation d'hypothèses existantes.
2. *Sauvegarde et mise à jour des cas potentiels sous forme de cas concrets.* Une fois identifiés, ces cas potentiels peuvent être enregistrés sous la forme de cas concrets. La description de ces cas peut alors être modifiée suivant les raisonnements. Nous pallions ainsi au manque des approches existantes (sauf SINS) qui extraient des cas ou des parties de cas, sans pouvoir retenir les conséquences de leur utilisation. La structure des situations permet alors de sélectionner et de modifier les indices effectifs parmi les indices potentiels : définition de comportements élémentaires et expression de contraintes temporelles. Ainsi, depuis les situations simples issues de l'instanciation des patrons, les raisonnements peuvent construire des connaissances plus précises et faire émerger des comportements plus complexes, proches de scénarios temporels. De plus, il est possible d'associer à chaque cas concrets des informations complémentaires (par exemple l'évaluation de la qualité du cas suivant un critère lié à l'application) qui seront utilisées lors des raisonnements.
3. *Interprétation facilitée des cas concrets.* Contrairement à SINS qui effectue un apprentissage en créant des cas virtuels, nous permettons une démarche visant à déterminer, sur une expérience concrète, les indices effectifs tout en éliminant les descriptions qui sont jugées non pertinentes (indices potentiels). Ainsi, les cas concrets référencent toujours les données brutes réelles pour faciliter leurs interprétations par des experts du domaine.

Notre modèle de représentation permet une expression riche des situations mais peut nécessiter une identification coûteuse : mise en correspondance de multiples comportements élémentaires et de contraintes temporelles, instanciation des patrons de cas potentiels. Nous donnons toutefois l'exemple d'une stratégie de recherche séquentielle combinant différents types de filtrage pour réduire progressivement les cas à examiner et retarder les étapes coûteuses. De plus, la transformation des cas potentiels en cas concrets, combinée avec la stratégie séquentielle de recherche proposée, amène une meilleure efficacité, puisque l'extraction de comportements n'est pas effectuée à chaque raisonnement.

Notre modèle d'indexation a été utilisé pour la spécification et la réalisation d'un prototype de recherche des cas pour la nutrition des plantes. Dans ce domaine, nous pensons que l'utilisateur est capable d'effectuer lui-même, lors de la phase d'apprentissage, la modification de l'indexation d'un cas source suite à son utilisation. Par exemple, si le comportement prévu par un cas source (concret ou potentiel) ne s'est pas reproduit, il peut analyser les indices et ajouter un comportement élémentaire qui pourrait expliquer cette différence. Chaque cas représente ainsi une hypothèse de l'explication d'un comportement, et le système entier se présente comme un gestionnaire d'hypothèses. La mise en place d'un tel système dans un environnement réel permettrait d'effectuer l'acquisition des connaissances pour concevoir à terme un système automatique.

Notre modèle a été également utilisé dans la réalisation complète et opérationnelle du système Broadway pour l'assistance à la navigation sur le Web (cf. chapitre V). Ce système permet d'évaluer les principales caractéristiques du modèle d'indexation : représentation des cas concrets et potentiels,

et guide d'utilisation pour les phases du raisonnement. La création de cas abstraits par généralisation de chroniques, ainsi que la modification automatique des indices des cas, constituent des perspectives intéressantes qui permettraient d'étendre nos travaux.

Enfin, pour l'utilisation concrète de notre modèle, nous avons conçu une plate-forme à objets facilitant son utilisation par la spécialisation et la composition d'un ensemble d'objets. Cette plate-forme facilite également l'intégration du modèle et son utilisation dans un système complet de raisonnement à partir de cas (cf. chapitres III et IV).