Chapitre I : le RàPC pour une classe de problèmes

Résumé

Notre étude a été motivée par deux applications : tout d'abord la régulation de la nutrition de plantes, puis l'assistance à la navigation sur le World-Wide-Web. Ces deux applications, bien qu'issues de domaines différents, partagent des contraintes communes, ce qui nous a amenés à considérer une *classe de problèmes* plus générale dans laquelle la prise en compte d'historiques est choisie ou nécessaire pour formuler le problème. Notre objectif est de faciliter l'utilisation du raisonnement à partir de cas pour cette classe de problèmes. Dans ce but, il est premièrement nécessaire de définir un *modèle générique d'indexation des cas* permettant la gestion d'historiques. Ce modèle doit fournir un formalisme générique de représentation des indices et doit permettre le développement d'une compétence méthodologique accumulée à travers l'analyse de différentes applications. Deuxièmement, il s'agit également de concevoir un *outil ouvert* facilitant l'utilisation pratique de ce modèle pour la réalisation de systèmes de raisonnement à partir de cas. Cet outil doit permettre la spécialisation du modèle pour prendre en compte les caractéristiques spécifiques d'une application. Il doit également faciliter son intégration cohérente dans un système complet de RàPC. Enfin, cet outil doit permettre la capitalisation de composants pour enrichir, au cours des développements, l'ensemble des objets-métier réutilisables.

Dans ce premier chapitre, nous décrivons tout d'abord le contexte des deux applications qui ont motivé notre étude : la régulation de la nutrition des plantes et l'assistance à la navigation sur le World-Wide-Web. La première application servira de cadre pour un premier exemple de notre approche dans le chapitre suivant. La deuxième application sera détaillée dans le chapitre V. Nous présentons ensuite les principes du raisonnement à partir de cas en introduisant notamment les étapes du raisonnement et les conditions de son application. Enfin, nous expliquons notre démarche et les objectifs qui ont guidé toute notre étude et la rédaction de la thèse.

1 Motivations

La classe de problèmes abordée dans cette thèse est principalement motivée par l'étude de deux applications : la régulation de la nutrition des plantes basée sur les travaux des chercheurs de l'INRA[2] de Sophia-Antipolis, et l'assistance à la navigation sur le World-Wide-Web. Nous présentons ces deux applications en montrant l'intérêt d'adopter une approche basée sur la réutilisation d'expériences passées.

1.1 Régulation de la nutrition des plantes

Nous donnons, dans un premier temps, le contexte général de la nutrition des plantes, puis nous présentons plus particulièrement le problème de régulation que nous avons étudié et l'approche que nous visons.

1.1.1 Problème de la nutrition

Dans le cadre donné par l'INRA (Brun *et al.,* 1993 ; Brun & Settenbrino, 1993), l'objectif général est de *réguler* la nutrition minérale et hydrique du rosier en culture hors sol de manière à maîtriser les dérives minérales dans le substrat de culture. Plus précisément, il s'agit d'assurer une meilleure adaptation de l'offre aux besoins de la plante (pour éviter les carences et les stress[3]) tout en limitant les rejets dans l'environnement. La régulation d'un processus dynamique, comme celui de la nutrition, consiste à déterminer les valeurs d'une ou plusieurs variables contrôlées du système de manière à optimiser un critère d'évaluation dépendant d'une *consigne*. La technique utilisée de culture hors sol sous serre permet, d'une part, de contrôler à l'aide d'ordinateurs certains paramètres (nutrition minérale et hydrique, température) et d'autre part, de recueillir régulièrement, grâce à des capteurs ou des mesures manuelles, un ensemble de données issues de la plante et de son environnement (par exemple le diamètre des tiges, la température, l'humidité, le rayonnement solaire, la concentration et le volume du drainage).

Ce problème général de régulation a été décomposé par les chercheurs de l'INRA en sous-problèmes à des échelles de temps différentes. Trois niveaux de régulation sont identifiés (Baille *et al.,* 1990 ; Baille, 1991) : contrôle du microclimat effectué presque de manière instantanée, contrôle de la croissance de la culture avec des décisions journalières, et contrôle de la croissance à long terme où l'unité de temps est le mois. Au premier niveau, nous notons deux régulateurs importants, l'un chargé de l'irrigation et l'autre de la température. Le premier régulateur permet de calculer le volume de la solution nutritive de manière à maintenir dans des limites fixées le volume du drainage (20 à

[2] Institut National de Recherche en Agronomie.
[3] Toutefois, un stress hydrique appliqué à un stade déterminé de la croissance est parfois favorable pour le rendement.

30% du volume de l'irrigation) : l'irrigation de doses de concentration constante est déclenchée en relation avec le rayonnement global. La température à l'intérieur de la serre est également régulée dans une certaine limite : un chauffage peut être déclenché la nuit lorsque la température passe en dessous d'un seuil (14º), et si la température dépasse un autre seuil durant la journée (26º), l'ouverture du toit de la serre et l'activation de la ventilation permettent de la faire baisser.

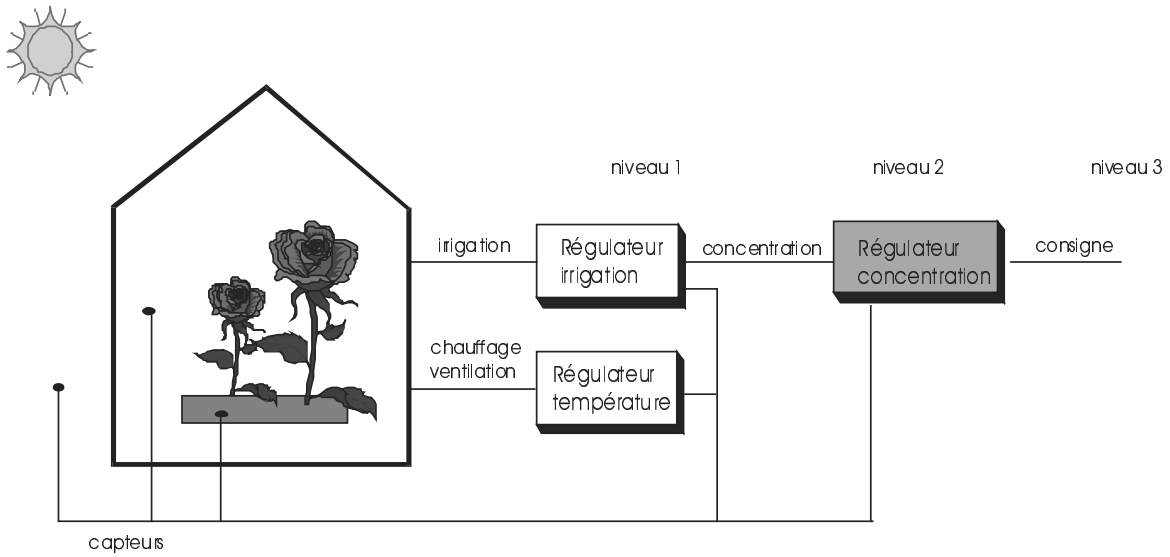

Figure I-1 : régulateur de concentration

Nous nous sommes intéressés plus particulièrement à un problème de régulation de la concentration journalière de drainage (Brun *et al.*, 1993 ; Brun & Settenbrino, 1993). Dans ce régulateur, situé au deuxième niveau (cf. Figure I-1), il s'agit de maintenir autour d'une consigne (déterminée au troisième niveau) la concentration en sels minéraux de la solution de drainage qui est récupérée après l'arrosage des plantes. La commande, c'est-à-dire le paramètre contrôlé, est la concentration de la solution d'irrigation. Les concentrations sont évaluées grâce à l'électro-conductivité des solutions ; la consigne est ainsi déterminée *a priori*, d'une part à partir des besoins estimés pour une bonne croissance de la plante qui donne une certaine électro-conductivité, et d'autre part à partir de l'électro-conductivité de l'eau servant à préparer la solution. Cette régulation est influencée par des perturbations dont certaines sont effectivement mesurées sur la base de la journée (température, rayonnement solaire, humidité).

1.1.2 Vers la réutilisation d'expériences de régulation

Pour mieux comprendre le fonctionnement du processus de nutrition, nous avons dressé les principales relations qualitatives entre les différents paramètres à partir des discussions que nous avons eues avec M. Brun, chercheur de l'INRA. La plante reçoit un rayonnement et produit de la vapeur d'eau par transpiration. Au niveau des racines, la plante absorbe de manière sélective de l'eau et différents sels minéraux. Ainsi, plus le rayonnement est fort, plus la transpiration est forte et la plante absorbe plus d'eau que de sels minéraux. Dans ce cas, la concentration de l'apport à l'intérieur de la plante est faible, « elle consomme plus dilué ». Par contre, si le rayonnement est faible, la transpiration est faible et le besoin en eau également, « la plante consomme plus concentré ». Connaissant ce phénomène, il est inutile de trop concentrer l'apport lors d'un rayonnement fort, car beaucoup de sels minéraux se retrouveraient dans le drainage. Les besoins en eau sont évalués par une formule prenant en compte le rayonnement et parfois l'humidité : plus le rayonnement est fort, plus l'espacement entre les doses est faible, et plus l'apport journalier est grand.

Le processus est également caractérisé par une inertie : la concentration de l'apport dans les racines ne peut pas être changée immédiatement. En effet, il s'écoule un certain temps entre le changement de concentration de l'irrigation et celle dans le milieu racinaire. Cette inertie peut être évaluée en mesurant le temps entre une modification importante de la concentration de l'irrigation qui correspond à un certain degré d'électro-conductivité et à sa réponse au niveau de l'électro-conductivité du drainage. Des expériences montrent que ce retard dépend de l'ordre de grandeur des concentrations mises en jeu et du volume de l'irrigation : plus la concentration est grande, plus le retard est grand, et d'un autre côté lorsque l'irrigation est plus forte (durant l'été), l'inertie baisse. Ainsi, cette inertie varie de 2 à 5 jours suivant les paramètres climatiques (Brun *et al.,* 1993). Il est donc important de prendre en compte les prévisions climatiques à court terme pour anticiper les effets de l'inertie dans la stratégie de régulation.

De manière générale, la régulation d'un système peut s'effectuer en boucle ouverte (*feedforward control*), en boucle fermée (*feedback control*) ou par une régulation mixte (*feedforward-feedback control*) (Box & Jenkins, 1976). En boucle ouverte, la régulation se base sur les mesures des perturbations de manière à ce que la commande annule leurs effets. La régulation en boucle fermée permet de prendre en considération les perturbations non mesurées qui influencent le fonctionnement du système. En effet, dans ce cas, la loi de commande du régulateur prend comme paramètre l'écart entre la consigne et la valeur réelle. La régulation mixte permet de tirer avantage des deux approches pour donner généralement de meilleurs résultats : les perturbations mesurées et non mesurées sont prises en compte par le régulateur. Les problèmes de régulation sont généralement résolus par des approches issues de l'Automatique où l'identification des modèles formels du procédé et des perturbations est nécessaire (Landau, 1988 ; Faurre & Robin, 1986). Ces approches permettent de garantir certaines caractéristiques de la commande (robustesse). Ainsi, une approche de régulation mixte, inspirée des régulateurs PID (actions Proportionnelles Intégrales et Dérivées), a été mise en place (Brun *et al.,* 1993 ; Brun & Settenbrino, 1993). Toutefois, les caractéristiques du processus de nutrition (inertie, perturbations, prise en compte des prévisions climatiques, nombre important de variables) et l'absence de modèles formels complets laissent ouvert ce problème de régulation pour l'application d'autres techniques, de manière à améliorer les résultats obtenus. C'est pourquoi, plus généralement, des approches issues de l'Intelligence Artificielle ont déjà été appliquées dans des contextes de nutrition de plantes : systèmes experts (Fynn *et al.*, 1989), techniques d'apprentissage (Kurata, 1988) et réutilisation d'expériences passées (Chiriatti & Plant, 1993).

Contrairement à ces approches, nous proposons de réutiliser des expériences passées concrètes de régulation en prenant en compte les *évolutions* passées de variables ainsi que leurs prévisions. Ces évolutions sont déterminantes dans la prise de décision pour la régulation notamment à cause du phénomène d'inertie. De plus, notre approche vise à réutiliser les expériences passées de régulation pour reproduire les séquences de commandes réussies, tout en évitant celles dont la qualité n'est pas suffisante. Dans un premier temps, un atelier de commande mettant en œuvre une coopération homme-machine pour opérer la régulation est nécessaire, et permettra la mise en évidence de comportements plus généraux du processus. Ces connaissances générales pourront ensuite permettre la mise en place d'une automatisation complète.

1.2 Aide à la navigation sur le World-Wide-Web

Le World-Wide-Web, ou plus communément le Web, est un gigantesque hypermédia dont nous présentons tout d'abord les principales caractéristiques. Puis, nous introduisons nos motivations pour la réalisation d'un assistant de navigation réutilisant les expériences d'un groupe d'utilisateurs.

1.2.1 Présentation du Web

Le World-Wide-Web est né des recherches débutées en 1989 au CERN par Tim Berners-Lee. L'objectif était de disposer d'un système permettant à un ensemble de chercheurs, répartis sur plusieurs sites, de pouvoir accéder et publier des informations de manière simple et unifiée, pour constituer une base documentaire partagée. Une contrainte majeure était de pouvoir relier des documents de projets *a priori* indépendants, si des liens pertinents étaient identifiés par la suite, sans avoir à effectuer des modifications lourdes ou la mise à jour d'une base centrale (Berners-Lee *et al.,* 1994). Le Web repose ainsi sur la notion d'*hypertexte* (cf. Figure I-2) : un hypertexte est un ensemble de nœuds contenant des informations textuelles dans lequel certaines expressions sont liées à d'autres nœuds. Ce type de document électronique permet une lecture non linéaire, et la consultation des documents se fait alors en « navigant » au travers des liens pour rechercher l'information voulue suivant son intérêt.

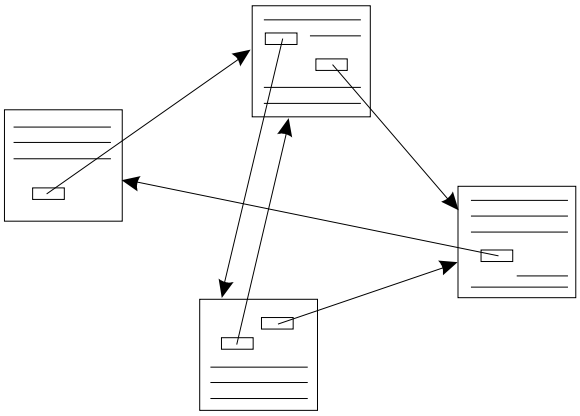

Figure I-2 : exemple simple d'hypertexte (Vanoirbeek, 1996)

Tim Berners-Lee définit le Web comme étant « un système d'information réparti, coopératif, hétérogène et multimédia »[4]. Le Web est en effet un système d'information réparti qui s'étend dans le monde entier puisque les serveurs de documents peuvent être physiquement localisés à différents endroits et que les liens entre les documents peuvent référencer indifféremment n'importe quel autre serveur. Le Web permet aussi de mettre en œuvre la constitution coopérative de documents, car chaque utilisateur peut mettre à jour ses propres données en les liant à celles des autres. Le Web est ensuite hétérogène puisqu'un même document peut être consulté depuis des ordinateurs et des plates-formes différents. Un utilisateur possède également un accès identique aux documents issus des différents serveurs bien que ces derniers soient d'architectures différentes. Le Web est enfin un multimédia puisqu'il permet de relier des documents de différents types : textes bien sûr mais aussi images, animations, sons... Le Web est donc plus qu'un hypertexte, c'est un *hypermédia*.

Les logiciels utilisés pour consulter le Web sont appelés des *navigateurs*, et la Figure I-3 donne l'exemple d'un navigateur affichant une page[5] Web (nœud de l'hypermédia) présentant les différents pointeurs (expressions soulignées) sur d'autres pages à propos de l'INRIA : il suffit alors de cliquer sur l'expression choisie pour automatiquement afficher la page correspondante qui peut se trouver physiquement sur le serveur de l'INRIA ou sur un autre serveur distant. Certains liens peuvent aussi référencer des sous-parties de la page courante.

[4] http://www.w3.org/Talks/General.html

[5] Nous préférons le terme de *page* plutôt que celui de *document* qui fait référence à « une unité de sens autonome et complète liée au médium papier » (Chartron, 1996). En fait, un document Web est très souvent constitué de plusieurs pages.

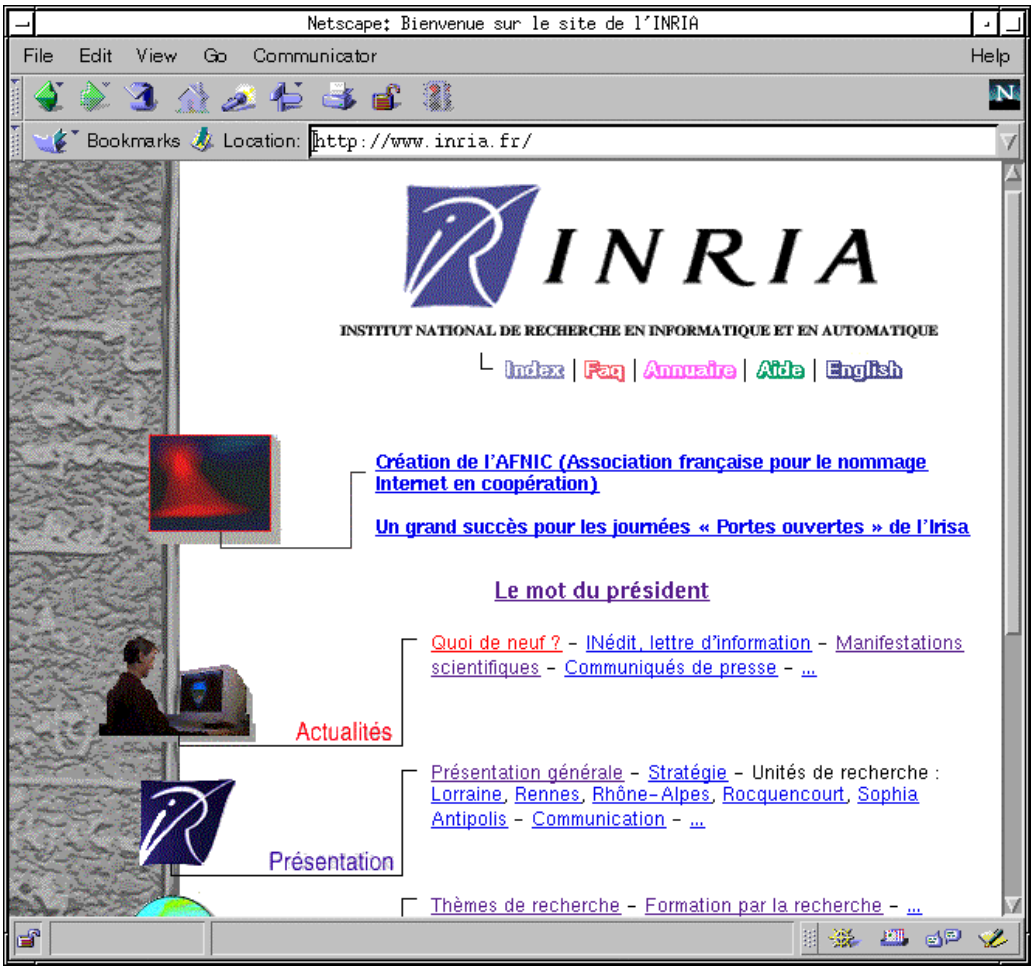

Figure I-3 : affichage d'une page Web avec le navigateur Netscape

Sur le plan de la communication entre les clients et les serveurs, le Web s'appuie sur Internet, une « interconnexion de réseaux couvrant la planète » (Huitema, 1995). Avec des débuts en 1969 par le réseau de l'armée américaine Arpanet, Internet regroupe un ensemble d'ordinateurs et de protocoles de communications permettant leur connexion (dont notamment les protocoles *Internet Protocol* et *Transmission Control Protocol*). Au-dessus de ces protocoles, un protocole spécifique, nommé HTTP (*Hypertext Transfer Protocol*), a été ajouté pour l'échange des pages hypertextes entre les serveurs du Web et les clients (RFC 2068, 1997). Toutefois, le Web ne se limite pas aux pages accessibles directement par ce protocole, mais définit un espace virtuel dans lequel chaque élément est accessible par une adresse appelée URI[6] (*Uniform Resource Identifier*), où l'utilisation d'autres protocoles existants est possible (RFC 1738, 1994). Cet espace est souvent défini par les éléments accessibles par les protocoles FTP (*File Transfer Protocol*) et Gopher, en plus du protocole HTTP (cf. Figure I-4). Cependant d'autres protocoles sont aussi utilisés : WAIS (*Wide Area Information Server*), Telnet, NNTP (*Network News Transport Protocol*).

Les différentes pages du Web sont généralement écrites dans un langage de formatage nommé HTML[7] (*Hypertext Mark-up Language*) permettant de standardiser l'affichage (Raggett, 1997 ; Raggett *et al.*, 1998). Certaines pages sont stockées dans les fichiers des serveurs et d'autres sont

[6] Pour des raisons historiques, URL (*Uniform Resource Locator*) est également très souvent employé. Toutefois, nous avons opté pour le nouveau terme URI (RFC 2068, 1997).
[7] Un autre langage, nommé XML (*eXtensible Mark-up Language*), est également à l'étude pour améliorer la gestion des documents, notamment en terme de structuration.

construites dynamiquement suivant le contexte (interrogation d'une base de données par exemple). Enfin, de nombreux formats sont également utilisés pour les documents non hypertexte : texte non formaté, Postscript, PDF (*Portable Document Format*)...

Bien que l'évaluation de l'étendue du Web soit une tâche difficile, certaines estimations (Bray, 1996) datées de 1995 montraient que le Web comportait 223 851 serveurs différents et 11 366 121 URI accessibles ; cependant les auteurs précisaient que des estimations plus récentes de 1996 portaient déjà ce dernier nombre à plus de 50 millions. Cet espace virtuel est en expansion croissante et offre ainsi une énorme quantité de données.

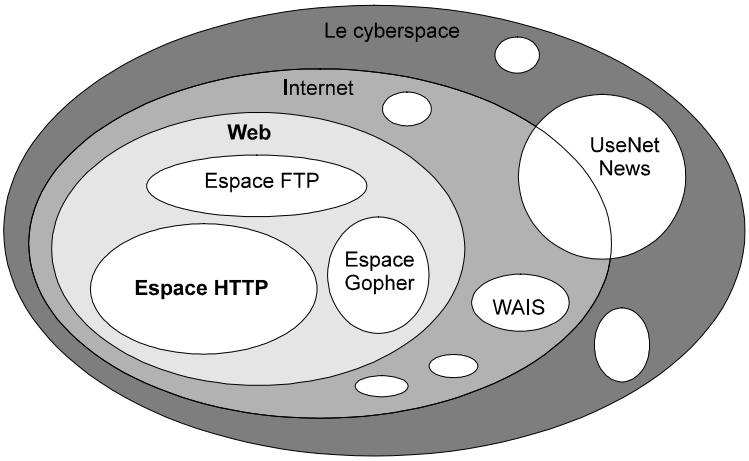

Figure I-4 : l'espace virtuel du Web pour Lycos[8]

1.2.2 Assistance à la navigation sur le Web

La recherche d'informations pertinentes sur le Web est une activité importante de l'utilisateur qui dispose d'un ensemble d'outils généraux ou spécialisés (Chartron, 1996). Les outils généraux regroupent : les répertoires thématiques construits en partie grâce à une intervention humaine (Yahoo[9], QuiQuoiOu[10] de France Télécom), les moteurs de recherche faisant une indexation automatique grâce à des robots qui parcourent le Web (Altavista[11], Lycos[12]), et les métaoutils permettant de combiner plusieurs autres outils (MetaCrawler[13], CBKB (Andreoli *et al.*, 1996) de Xerox). Les outils spécialisés utilisent des techniques similaires mais restent liés à un pays, à un espace linguistique ou à un type de contenu. Par exemple, les pages regroupant un ensemble de liens pertinents établis par un spécialiste constituent des outils très spécialisés.

Du point de vue de l'utilisateur, ces outils proposent deux types d'interaction : la *formulation de requêtes* (moteurs de recherche et métaoutils existants) et la *navigation hypertextuelle* (répertoires thématiques, pages spécialisées, serveurs Web). Cependant, la navigation reste le moyen principal de recherche. En effet, les moteurs de recherche nécessitent l'expression d'une requête généralement sous la forme de mots clefs. Cette requête peut être difficile à formuler si l'utilisateur n'a pas un but très précis. De plus, les résultats obtenus présentent un bruit important provenant de l'indexation automatique utilisée. C'est pourquoi l'utilisateur est très souvent obligé de naviguer, les résultats des outils servant tout de même de points de départ raisonnables (Jerke *et al.*, 1990). Toutefois, la

[8] Lycos est un robot de recherche indexant les informations disponibles par les protocoles HTTP, Gopher et FTP (cf. http://www.vperson.com/mlm/lycos-websize-9510.html).
[9] http://www.yahoo.com
[10] http://www.wanadoo.fr/bin/frame.cgi?service=quiquoiou
[11] http://altavista.digital.com/
[12] http://www.lycos.fr/
[13] http://www.metacrawler.com/

navigation est également une tâche difficile. Dans les répertoires, le facteur humain induit des biais dans la réalisation de la classification et ne permet pas d'être aussi exhaustif qu'une indexation automatique par robots. De plus, la complexité et l'étendue du Web peuvent entraîner la désorientation de l'utilisateur lors d'une navigation. Ce dernier problème est d'ailleurs général aux systèmes hypertextes (Foss, 1989; Dillon *et al.*, 1990).

Ainsi, la navigation hypertextuelle est nécessaire pour la recherche d'information sur le Web, et reste difficile. Pour faciliter cette navigation, une première approche est d'améliorer la conception des pages Web en définissant des règles pour les auteurs (Bachiochi *et al.,* 1997), mais ces dernières seront de toute manière plus ou moins suivies. C'est pourquoi, dans notre travail, nous nous orientons vers des systèmes de recommandations comme WebWatcher (Armstrong *et al.*, 1997) et Letizia (Lieberman, 1995), dans lesquels l'utilisateur est observé, et des recommandations de pages pertinentes lui sont faites. S'il juge ces recommandations intéressantes, l'utilisateur peut alors suivre les liens proposés. Ces approches permettent de guider l'utilisateur durant sa navigation.

Plus précisément, nous visons un assistant permettant une recherche d'information coopérative, où l'expérience d'un groupe de personnes est mise à profit. La prise en compte du *comportement des utilisateurs* (cf. Figure I-5), constitué notamment de la séquence des pages visitées et de leurs contenus, doit également permettre de mieux cerner l'intérêt de l'utilisateur pour affiner les recommandations. Notre approche soutient l'hypothèse suivante : *si deux utilisateurs ont navigué suivant une séquence similaire de pages, c'est qu'ils ont des intérêts similaires, nous pouvons alors proposer au second utilisateur les pages jugées pertinentes par le premier*. Ainsi, cet assistant doit permettre la réutilisation de navigations passées d'un groupe d'utilisateurs pour réduire la longueur des navigations et faciliter ainsi la recherche d'information.

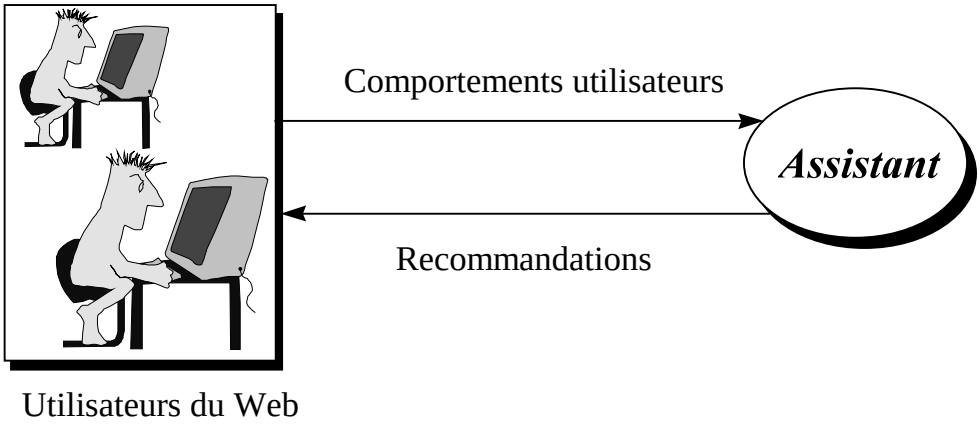

Figure I-5 : principe d'un assistant à la navigation

2 Présentation du raisonnement à partir de cas

Après avoir décrit nos motivations, nous introduisons les notions importantes du raisonnement à partir de cas. De manière générale, le raisonnement à partir de cas est une approche de résolution de problèmes basée sur la réutilisation par analogie d'expériences passées appelées *cas* au cours d'un cycle de raisonnement. Nous présentons tout d'abord les principes de ce type de raisonnement puis nous donnons les caractéristiques de son utilisation. Nous ne donnons pas dans cette section une

description exhaustive du raisonnement à partir de cas, notre objectif étant uniquement de présenter les concepts et la problématique de cette approche pour introduire le vocabulaire utilisé par la suite[14].

2.1 Principes du raisonnement

Le raisonnement à partir de cas est une approche de résolution de problèmes datant du début des années 1980 et issue de travaux en Sciences Cognitives et en Intelligence Artificielle (Schank, 1982 ; Kolodner, 1983 ; Riesbeck & Schank, 1989 ; Kolodner & Leake, 1996). Bien que le raisonnement à partir de cas soit inspiré de certaines observations sur le raisonnement humain, le RàPC prend aussi en compte le besoin pragmatique de techniques et de processus opérationnels.

2.1.1 Représentation des cas et des indices

Schématiquement, un cas est composé de deux parties décrivant : un *problème* et la *solution* qui lui a été appliquée. Sous ces deux termes simples, une représentation complexe est souvent nécessaire pour tirer parti des enseignements d'un cas de manière opérationnelle. La description du problème peut recouvrir les objectifs à atteindre, les contraintes sur ces objectifs et le contexte dans lequel le problème a été posé. La solution peut regrouper la description de la solution effectivement apportée, les étapes qui ont mené à cette solution, son évaluation et les effets de son application. Cette représentation dépend de la tâche à accomplir, du type des données disponibles et des connaissances du domaine. Bien que le raisonnement à partir de cas repose sur la réutilisation d'enseignements spécifiques, des cas abstraits (prototypes de cas) peuvent être définis (Bergman & Wilke, 1996). Cependant ces représentations plus générales doivent rester à un niveau *opérationnel*, c'est-à-dire que leur utilisation doit être uniquement basée sur des mécanismes d'instanciation et d'adaptation (Kolodner, 1996).

Idéalement, un cas est une expérience passée permettant au système de raisonnement à partir de cas de résoudre des problèmes plus efficacement ou d'éviter les échecs passés : un cas doit donc être *utile* pour le raisonnement (Kolodner, 1993). Dans ce but, un ensemble de caractéristiques est extrait ou sélectionné dans la représentation de chaque cas. Ces caractéristiques, appelées *indices*, déterminent dans quelle *situation* un cas est applicable et utile. Les indices doivent tout d'abord être prédictifs, c'est-à-dire qu'ils doivent être responsables de l'élaboration de la solution mise en œuvre dans le cas et permettront ainsi de déterminer dans quelle mesure une solution est applicable pour un autre problème. Ils doivent également être suffisamment abstraits pour permettre la réutilisation des enseignements du cas pour différents problèmes, tout en restant suffisamment concrets pour être facilement identifiables dans un nouveau problème.

2.1.2 Cycle du raisonnement

Lorsqu'un *nouveau problème* est à résoudre, il est intégré dans un *cas cible*[15] où la partie solution est inconnue et doit être apportée par le raisonnement. Les *cas sources* représentent des expériences passées qui sont stockées dans une *mémoire*. Les cas sources et les cas cibles partagent le même formalisme de représentation. L'objectif du raisonnement est de transférer des enseignements

[14] Pour plus de renseignements sur le raisonnement à partir de cas, le lecteur pourra se reporter aux ouvrages et publications spécialisées (Gebhardt *et al.,* 1997 ; Watson, 1997 ; Aamodt & Plaza, 1994 ; Kolodner, 1993 ; Slade, 1991 ; Kolodner & Leake, 1996).

[15] Le terme « nouveau cas » est parfois employé (Aamodt & Plaza, 1994). Nous préférons le terme de « cas cible » qui convient mieux à l'objectif du raisonnement où il s'agit de transférer des connaissances depuis des anciens cas, les cas sources, vers un cas courant, le cas cible. De plus, le terme « nouveau cas » peut induire en erreur, en suggérant que ce cas sera obligatoirement ajouté à la mémoire, or seule la phase d'apprentissage décidera en fin de raisonnement si l'expérience courante doit effectivement devenir un nouveau cas de la mémoire.

pertinents des cas sources pour élaborer la solution du cas cible. Ce raisonnement se décompose habituellement en quatre phases principales (Aamodt & Plaza, 1994) qui peuvent être totalement ou partiellement automatisées (cf. Figure I-6) :

1. phase de *recherche* dont le but est de sélectionner un ou plusieurs cas utiles pour la suite du raisonnement, tout en prenant en compte les indices du cas cible déterminés par une étape d'élaboration ;
2. phase de *réutilisation* permettant de construire une solution au problème courant en se basant sur les cas identifiés dans la phase précédente ;
3. phase de *révision* de la solution qui permet d'affiner la solution grâce à son évaluation ;
4. phase *d'apprentissage* chargée de mettre à jour les éléments du raisonnement en prenant en compte l'expérience qui vient d'être réalisée, et qui pourra ainsi être utilisée dans les raisonnements futurs.

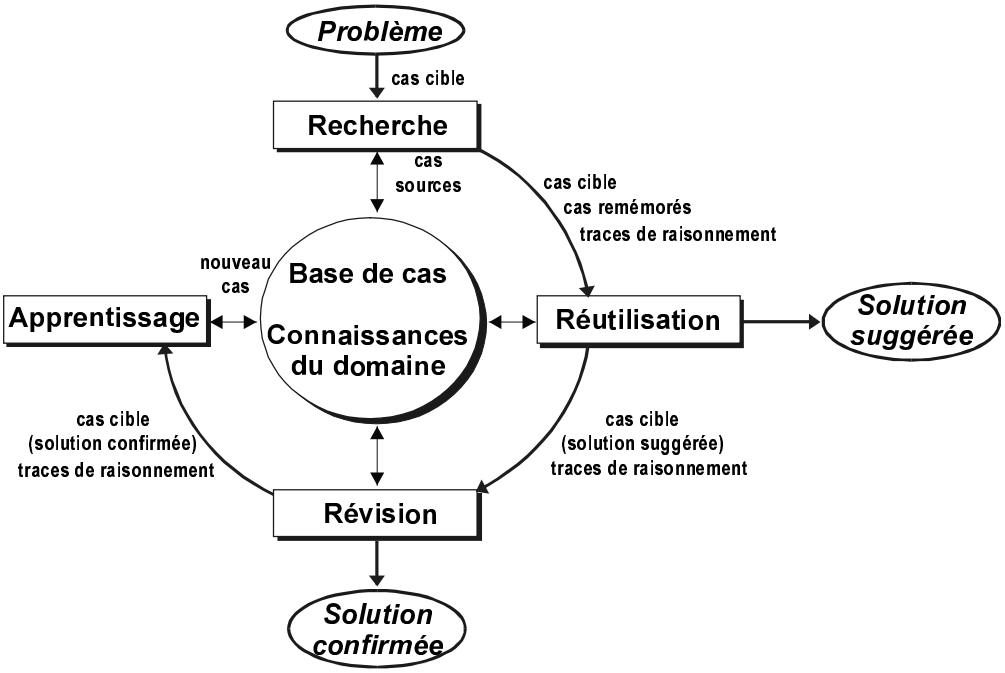

Figure I-6 : cycle du raisonnement à partir de cas

La problématique de la phase de recherche est de permettre d'identifier un ou plusieurs cas utiles pour le raisonnement en cours. La mesure de cette utilité peut être basée sur la similarité entre les indices du cas cible et des cas sources, ou sur des critères d'adaptabilité (Leake *et al.*, 1997). Généralement, deux étapes sont réalisées : *filtrage* des cas sur des indices de surface, puis *sélection* fine des cas. Toutefois, nous pouvons considérer l'étape de sélection comme un filtrage particulier, et en généralisant, nous dirons que la phase de recherche se compose d'une succession de filtrages. La réalisation de ces filtrages s'appuie sur la notion d'index qui permet d'organiser les cas sources au sein de la mémoire, en utilisant leurs indices. Un index est une structure de données utilisée par un algorithme offrant une recherche efficace, tout en utilisant des connaissances du domaine et/ou des connaissances induites pour identifier les cas les plus utiles. Dans ce sens, l'objectif d'un index dépasse les approches classiques de recherche des bases de données, même si des techniques issues de ce domaine sont parfois utilisées. Par exemple, un *index par plus proches voisins* s'appuie sur une structure linéaire des cas et sur une mesure de similarité. Dans cet index, la similarité entre le cas cible et chaque cas source est calculée, pour ne retenir que les k meilleurs (algorithme en $O(n)$ où n est le nombre de cas sources). D'autres index sont couramment utilisés (cf. chapitre III) : index par arbre d'induction ou index par hiérarchie de prototypes.

Etant donné qu'il est peu probable de retrouver un cas correspondant exactement au problème courant, la phase de réutilisation permet de construire une solution par adaptation des solutions passées des cas sources identifiés. Cette réutilisation se base principalement sur les différences entre les cas sources et le cas cible pour déclencher des méthodes d'adaptation. Après adaptation, la solution suggérée au problème courant peut être donnée à l'utilisateur uniquement si la phase de révision n'est pas effectuée ou si elle est manuelle.

La phase de révision permet l'évaluation et la réparation de la solution déterminée à la phase de réutilisation. Cette évaluation peut être basée sur la vérification de la solution par un expert, par une simulation, ou même par une exécution réelle. Si la solution est mauvaise, des explications peuvent être fournies ou générées pour que le système (ou un opérateur) puisse effectuer des actions de réparation afin d'améliorer la solution et de fournir la solution confirmée.

Dans la phase d'apprentissage, le système doit faire la synthèse du raisonnement qui a été suivi de manière à sauvegarder les nouveaux enseignements pertinents qui permettront d'améliorer le raisonnement par la suite. Plusieurs actions peuvent alors être effectuées : indexation du cas cible dans la mémoire, amélioration et modification de l'indexation des cas existants dans la mémoire (par exemple création de prototypes regroupant un ensemble de cas similaires). Si le cas cible est jugé intéressant et qu'il doit être indexé dans la mémoire, la méthode d'apprentissage doit déterminer quels sont les indices du cas cible de manière à définir dans quelles situations le cas sera de nouveau utile pour le raisonnement.

2.1.3 Importance de l'indexation pour le raisonnement et terminologie

L'indexation en RàPC consiste à définir les indices d'un cas, c'est-à-dire les caractéristiques d'un cas qui indiquent ses conditions d'application et d'utilité (les indices). Cette tâche peut être appelée le « problème de l'accessibilité » (Kolodner, 1996, page 355) étant donné que la phase de recherche est effectuée en premier dans le cycle du raisonnement et que les indices doivent mener, dans cette phase, à l'identification des cas à utiliser par la suite. Cependant, le but du raisonnement est clairement de réutiliser les connaissances issues des cas, soit manuellement soit automatiquement. C'est pourquoi les indices d'un cas doivent définir la situation dans laquelle ses enseignements peuvent être réutilisés : un cas de succès est réutilisé pour atteindre les objectifs courants, et un cas d'échec est réutilisé pour prévenir un échec potentiel. Ainsi, le problème de l'indexation doit aborder plus que la problématique de l'accessibilité et nous pouvons le nommer *problème de la réutilisabilité*.

Les cas sources et le cas cible courant partagent un même formalisme de représentation, généralement formé d'attributs, pour que les opérateurs de mise en correspondance et d'adaptation puissent être appliqués. Dans ce formalisme, trois types d'attributs sont identifiés en fonction de leur rôle pour l'indexation :

- *attributs toujours ignorés* : par exemple dans le cadre d'une application donnée, le nom du cas et sa date de création peuvent ne jamais être pertinents comme indices,
- *indices potentiels* : ces indices, provenant principalement de la description d'un problème, peuvent être utilisés pour déterminer l'utilité d'un cas et ses conditions d'application,
- *indices effectifs* ou simplement indices : ces indices déterminent précisément, pour un cas donné, les indices potentiels qui sont effectivement sélectionnés pour guider l'utilisation du cas (situation du cas). Les indices effectifs d'un cas source peuvent être déterminés *a priori* ou en fonction du cas cible courant.

Durant le raisonnement, un ou plusieurs cas sources sont réutilisés pour calculer la solution du cas cible, et les indices interviennent dans les phases de recherche, de réutilisation et d'apprentissage. La phase de révision ne fait généralement pas appel aux indices dans la mesure où les méthodes de réparation se basent uniquement sur la solution qui vient d'être construite. Durant la phase de

recherche, un sous-ensemble des indices effectifs des cas sources et du cas cible, que nous appelons *indices de recherche*, est mis en correspondance pour identifier les cas à utiliser dans la suite du raisonnement. Puis, dans la phase de réutilisation, certains indices effectifs, les *indices d'adaptation*, sont comparés entre le cas source et le cas cible pour participer au choix des actions d'adaptation à effectuer. En effet, un des aspects de la réutilisation est d'analyser « les différences entre le cas passé et le cas courant » (Aamodt & Plaza, 1994). Des indices d'adaptation peuvent ne pas être des indices de recherche : ils permettent une analyse plus fine des différences entre un cas source et un cas cible mais ne sont pas suffisamment discriminants pour être utilisés dans la recherche. Enfin, durant l'apprentissage, la mémoire peut être modifiée par l'ajout d'un cas ou la réorganisation de l'indexation grâce à la modification des indices effectifs au sein des indices potentiels et de leurs poids d'importance. Ainsi, le problème de l'indexation est au cœur du raisonnement à partir de cas et intervient dans les trois phases importantes du raisonnement (recherche, réutilisation, apprentissage).

Enfin, nous résumons la définition des termes introduits pour l'indexation, qui seront utilisés tout au long du document[16] :

- *Indice*. Un indice (effectif) est une caractéristique d'un cas permettant de définir ses conditions d'applicabilité et d'utilité.
- *Situation*. Une situation définit l'ensemble des indices effectifs d'un cas.
- *Filtrage*. La phase de recherche est décomposée au niveau connaissance en étapes de filtrage. Un filtrage permet de retourner un sous-ensemble de cas sources applicables et jugés utiles pour la suite du raisonnement. La dernière étape de filtrage est appelée sélection.
- *Index*. Un index est une structure de données utilisée par un algorithme permettant la réalisation effective d'un ou de plusieurs filtrages, tout en prenant en compte les connaissances du domaine et les contraintes de performance. Un index est également mis à jour lors de l'apprentissage.

2.2 Utilisation du raisonnement à partir de cas

Comme toute méthode de résolution de problèmes, le raisonnement à partir de cas présente des avantages, des inconvénients et des conditions d'application que nous décrivons brièvement. Nous donnons alors l'exemple d'un système de RàPC, nommé CLAVIER (Mark *et al.*, 1996), pour illustrer le cycle du raisonnement et sa problématique.

2.2.1 Avantages, inconvénients et conditions d'application

Le raisonnement à partir de cas présente de nombreux atouts et avancées dans le domaine de l'Intelligence Artificielle (Leake, 1996 ; Slade, 1991 ; Kolodner, 1992) portant notamment sur : l'acquisition des connaissances, la maintenance de ces connaissances, l'efficacité et la qualité du raisonnement et enfin son explication. Tout d'abord, sur le plan de l'acquisition des connaissances, le raisonnement à partir de cas demande généralement moins d'efforts que la constitution d'une base de règles. D'une part, dans certaines applications, les personnes raisonnent explicitement en utilisant des expériences passées et des dossiers existants peuvent être utilisés (Mark *et al.,* 1996). D'autre part, les cas représentent des enseignements spécifiques plus faciles à expliciter que des règles générales.

[16] La définition de ces termes est importante car il n'existe pas de consensus au sein de la communauté RàPC. Par exemple, le terme « index » peut représenter, suivant les auteurs, ce que nous appelons une situation ou encore un indice. De plus, en anglais les termes « *indices* », « *indexes* » et « *index* » ont la même signification, ce qui amène parfois une confusion regrettable.

Le raisonnement à partir de cas offre également des possibilités de maintenance de la connaissance puisque de nouveaux cas peuvent être ajoutés à la mémoire ou bien retirés (Smyth & Keane, 1995). Par exemple, un ensemble de cas doit être initialement conçu puis, suivant les besoins et les échecs du raisonnement, de nouveaux cas peuvent venir enrichir les connaissances du système.

De plus, l'utilisation du raisonnement à partir de cas peut permettre une plus grande efficacité du processus de résolution de problèmes (Veloso, 1996) car celui-ci n'est pas entièrement répété à chaque fois à partir des principes de base lorsqu'un problème similaire se présente. Dans les systèmes experts, par exemple, le même ensemble de règles doit être déclenché. La qualité du raisonnement est également améliorée, d'une part, grâce aux échecs passés qui peuvent être évités et, d'autre part, grâce à l'utilisation d'expériences passées réelles qui offrent une validité au raisonnement dans des domaines où les connaissances sont réduites.

Enfin, l'explication du raisonnement est générée beaucoup plus facilement et de manière plus satisfaisante que dans les systèmes experts ou d'autres systèmes apprentis comme les réseaux de neurones : les cas sources qui sont généralement des expériences vécues servent de justification. De plus, le fonctionnement du raisonnement étant intuitivement compréhensible par les utilisateurs, ces justifications sont d'autant mieux acceptées, et les utilisateurs peuvent être associés au raisonnement dans des systèmes interactifs (Mark *et al.,* 1996).

Toutefois, cette technique présente quelques inconvénients et des conditions d'application. Le problème majeur est l'application aveugle des expériences passées : notamment l'utilisation de mécanismes d'adaptation peut rendre obscures les justifications issues des cas sources aux yeux des utilisateurs (Mark *et al.,* 1996). De plus, le système peut ne pas proposer la solution optimale, puisque l'espace des problèmes peut ne pas être entièrement connu : ceci pose le problème de la *couverture du domaine*. L'évaluation de cette couverture est difficile et ne peut pas être réduite à un critère simple comme la taille de la base de cas (Kolodner, 1996). Le raisonnement à partir de cas doit être considéré lorsqu'il est difficile de formuler des règles et que des cas peuvent être représentés ou sont disponibles. Toutefois, l'application de ce type de raisonnement repose sur des hypothèses sur le modèle de l'univers en cours d'étude (Kolodner, 1996) :

- *Régularité* : les mêmes actions menées dans les mêmes conditions ont des effets identiques ou proches.
- *Typicalité* : les problèmes se répètent et les expériences apprises peuvent être utiles dans le futur.
- *Cohérence* : de petits changements dans l'univers nécessitent seulement de petits changements dans la façon dont on l'interprète et dans les solutions.
- *Facilité d'adaptation* : bien que tout ne se répète pas à l'identique, les différences tendent à être petites, et ces petites différences peuvent être facilement prises en compte.

Très souvent une modélisation trop simpliste de cet univers ne permet pas de garantir ces hypothèses. Il s'agit donc premièrement de modéliser cet univers en choisissant une représentation suffisamment *abstraite* pour les cas et notamment pour les indices. Ainsi, même si des *caractéristiques de surface* peuvent facilement être identifiées, il s'agit de vérifier leur qualité en tant qu'indices et notamment leur prédictivité, sinon des caractéristiques plus abstraites sont nécessaires (Kolodner, 1996). Deuxièmement, les traitements effectués pendant le raisonnement sont déterminants. Par exemple, au lieu de réutiliser directement une ancienne solution par transformation, *l'analogie par dérivation* (Carbonell, 1986) propose de réutiliser les étapes qui ont mené à l'élaboration de cette solution. Enfin, dans un domaine où les experts ne raisonnement pas eux-mêmes en réutilisant des expériences passées, toutes ces hypothèses sont difficilement vérifiables *a priori* et devront généralement être évaluées par la réalisation et l'évaluation expérimentale d'un prototype.

2.2.2 Un exemple : le système CLAVIER

Le système de raisonnement à partir de cas CLAVIER est en utilisation journalière depuis les années 1990 chez Lockheed et permet la configuration d'un autoclave (Mark *et al.*, 1996 ; Althoff *et al.*, 1995, chapitre 6 ; Hennessy & Hinkle, 1992). Un autoclave est une sorte de four nécessaire à la fabrication d'éléments composites qui sont utilisés dans les avions et les missiles (cf. Figure I-7). Un autoclave est une ressource coûteuse, c'est pourquoi plusieurs éléments sont placés ensemble pour une même cuisson. Cependant, le résultat de la cuisson dépend grandement de la façon de positionner les éléments entre eux. Un seul défaut dans les pièces obtenues entraîne leur rejet et des pertes financières considérables. Les connaissances sur ce processus sont très limitées et une tentative de modélisation par règles avait d'ailleurs échoué. Comme les opérateurs utilisaient déjà des descriptions sous forme de fiches proches de la notion d'un cas, l'utilisation du raisonnement à partir de cas dans CLAVIER fut alors naturelle.

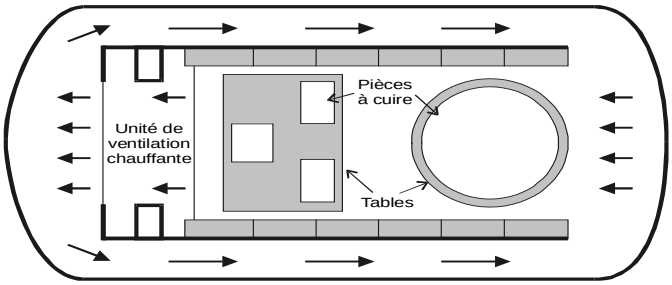

Figure I-7 : chargement de l'autoclave

Le système CLAVIER permet d'automatiser partiellement le positionnement des éléments dans l'autoclave. Un cas source représente une expérience passée donnant d'une part une liste d'éléments avec leur type et, d'autre part le positionnement utilisé dans l'autoclave et son résultat (valide ou invalide). Un cas cible est constitué d'une liste d'éléments à cuire avec leurs priorités et du positionnement à déterminer. Le raisonnement suit alors les phases suivantes :

- *Recherche*. La phase de recherche permet d'identifier le plus proche cas source valide proposant un positionnement pour le plus grand nombre d'éléments courants ayant les plus fortes priorités.
- *Réutilisation*. Ensuite, la phase de réutilisation permet d'adapter le positionnement en substituant les éléments utilisés dans le cas source et non mis en correspondance, par des éléments du cas cible qui n'ont pas encore été disposés. Cette adaptation peut être automatique ou manuelle. En fait, l'adaptation était au départ automatisée mais fut mal acceptée par les utilisateurs qui préféraient effectuer l'adaptation eux-mêmes.
- *Révision*. Dans ce dernier cas, le système vérifie alors la solution dans une phase de révision pour éviter de tomber sur un positionnement invalidé dans le passé. La révision est alors dans tous les cas finalisée par la mise en place de la disposition dans l'autoclave. A la suite de l'utilisation de l'autoclave, la disposition est validée ou invalidée.
- *Apprentissage*. Enfin, le cas cible courant est ajouté dans la mémoire qu'il soit valide ou invalide.

Dans cet exemple, les différents ensembles d'indices (indices potentiels et effectifs ainsi que les indices de recherche et d'adaptation) introduits précédemment, sont tous identiques : ils donnent la liste des éléments et leur type. Le type des éléments a en effet été jugé suffisant pour guider les phases du raisonnement ; les informations sur la géométrie ou la constitution des éléments sont ignorées.

3 Application du RàPC à une classe de problèmes

L'étude des problèmes de la régulation de la nutrition des plantes et de l'assistance à la navigation nous a permis de mettre en évidence une classe de problèmes dans laquelle l'observation d'un processus dynamique et la prise en compte d'historiques sont importantes. L'application du raisonnement à partir de cas dans ce contexte est intéressante et présente une spécificité notamment liée à la gestion d'indices reposant sur des historiques. Pour aborder cette classe de problèmes avec du RàPC, nous visons la définition d'un modèle générique d'indexation et la conception d'un outil ouvert facilitant son utilisation.

3.1 Classe des problèmes nécessitant la prise en compte d'historiques

Bien que les applications ayant motivé notre étude, la régulation de la nutrition des plantes et l'assistance à la navigation sur le Web, soient des problèmes issus de domaines bien différents, nous avons identifié certaines caractéristiques pertinentes et communes. Nous avons ainsi défini une classe de problèmes ayant les cinq propriétés suivantes :

1. La prise en compte *d'historiques* est choisie ou nécessaire pour formuler le problème.
2. Un *processus dynamique*, c'est-à-dire un processus évoluant dans le temps, est observé.
3. Un ensemble *important* de données brutes est collecté.
4. Les données brutes peuvent être obtenues par *échantillonnage* et/ou lors *d'événements*.
5. Les connaissances du domaine sont *réduites* et une meilleure compréhension est recherchée.

Nous pouvons alors vérifier que les approches prises dans les applications précédemment décrites font partie de cette classe de problèmes. Pour la nutrition, le processus est constitué par la boucle de régulation opérant sur des plantes dans un environnement. Pour la navigation sur le Web, le processus observé intègre les utilisateurs qui naviguent sur le Web avec l'aide de l'assistant. Il est à noter que, dans les deux cas, le processus observé comprend le système chargé de résoudre le problème posé : le régulateur ou l'assistant. En effet, ces systèmes sont actifs, modifient le comportement global du processus, et peuvent avoir des variables d'état dont les valeurs évoluent dans le temps. Dans les deux cas, un nombre important de données est collecté (perturbations, commandes, pages visitées, caractéristiques des documents...). Ces données peuvent être échantillonnées avec une période donnée (par exemple le jour), ou obtenues lors d'événements (déclenchement de la ventilation, alarme, visualisation de pages spécifiques). Dans les deux domaines, peu de connaissances formelles sont disponibles et la prise en compte d'historiques permet une meilleure description du problème à résoudre qu'une description par un ensemble figé de valeurs instantanées de variables d'état. Pour la nutrition, l'inertie du système implique la prise en compte des commandes passées et des évolutions des perturbations pouvant remonter à plusieurs jours. De même, pour la navigation, une meilleure description de l'intérêt de l'utilisateur lors d'une recherche d'informations est obtenue en prenant en compte les événements pertinents de l'historique de ses déplacements.

Enfin, l'intérêt d'aborder cette classe de problèmes dépasse le cadre de nos deux applications. Pour de nombreux problèmes, une composante essentielle réside dans l'observation d'un processus avec la prise en compte d'historiques. De manière générale en médecine, l'évolution de l'état d'un patient est souvent plus pertinente qu'un état particulier (Keravnou, 1995), et les fiches médicales ou les enregistrements doivent être consultés. Dans les problèmes d'analyse et de compréhension du comportement d'un être humain utilisant une machine, il est nécessaire de prendre en compte des séquences temporelles d'interactions (Ritter & Larkin, 1994). Dans le domaine de la conduite de procédés, des délais de réaction importants et variables peuvent être observés, ce qui oblige à prendre

en compte des événements passés de plusieurs minutes et même de plusieurs heures : par exemple, dans la conduite de hauts fourneaux, une erreur d'alimentation peut être observée six heures plus tard (Jakob & Suslenschi, 1990).

3.2 Intérêt et spécificité du RàPC

L'application du raisonnement à partir de cas pour cette classe de problèmes soulève une certaine spécificité dont l'impact majeur repose sur la représentation et la gestion des indices durant les raisonnements. Il s'agit en effet de représenter des historiques comme indices, tout en gardant une représentation compréhensible des indices permettant d'utiliser les cas comme explication du raisonnement.

3.2.1 Justification et intérêt du raisonnement à partir de cas

Nous abordons ainsi des problèmes liés à un processus dynamique pour lequel de nombreux paramètres peuvent être identifiés et observés, mais dont leurs influences sur le comportement du processus sont mal connues. Dans ce cadre, le rôle d'une approche basée sur le raisonnement à partir de cas est double :

- raisonner pour élaborer des solutions aux problèmes à résoudre (prévision, diagnostic, commande, assistance),
- mettre en évidence de nouvelles connaissances.

Le raisonnement à partir de cas est justifié puisque les connaissances du domaine sont réduites (absence de règles) et que l'acquisition des données brutes, nécessaires à la représentation des cas, ne pose en général aucune difficulté (présence de capteurs). L'identification des enseignements utiles issus de la masse de données brutes collectées ainsi que leurs réutilisations sont les points clefs d'une telle approche.

Le raisonnement à partir de cas est également adapté pour la mise en évidence de nouvelles connaissances en utilisant les cas comme explications et justifications du comportement du processus. Cet intérêt peut être illustré dans le cadre des deux applications ayant motivé notre étude. Pour la régulation de la nutrition des plantes, notre objectif est de réaliser un système intégrant l'utilisateur notamment dans la phase d'apprentissage pour l'enrichissement et la modification contrôlés de l'indexation. Ce processus permet de faire émerger des connaissances à propos de l'influence des paramètres sur l'évolution de la concentration du drainage, en utilisant les cas sources comme explications. Pour l'assistance à la navigation, l'identification de parcours typiques est capitale pour l'amélioration de la conception des documents mis à disposition sur le Web (pour des objectifs commerciaux par exemple).

3.2.2 Gestion d'indices sous forme d'historiques

Comme l'énoncé du problème à résoudre comprend des historiques, le formalisme de représentation des indices, nécessaire à la représentation des cas cible et sources, doit être adapté et permettre la représentation de séquences de valeurs pertinentes. Nous faisons ainsi une distinction forte entre d'une part une *situation instantanée*, ensemble d'indices effectifs définissant un état à un instant particulier et d'autre part, une *situation comportementale*, ensemble d'indices effectifs décrivant une évolution dans le temps de cet état. L'utilisation de situations instantanées est l'approche la plus répandue dans les systèmes de raisonnement à partir de cas : par exemple les

systèmes classiques[17] CHEF (Hammond, 1986), JULIA (Hinrichs, 1989) et CASEY (Koton, 1989) utilisent des situations instantanées. Le système CHEF permet de dériver des recettes suivant notamment les ingrédients et le mode de cuisson désirés en se basant sur d'autres recettes stockées dans des cas. Les cas sont indexés suivant un ensemble de buts satisfaits ou d'échecs à éviter. Les indices utilisés décrivent uniquement les caractéristiques pertinentes à l'instant où le cas a été appris : il s'agit d'une situation instantanée. De même, dans le système CLAVIER que nous avons brièvement présenté, les indices décrivent encore une situation instantanée en donnant une liste d'éléments. Ainsi, l'application du RàPC à notre classe de problèmes diffère des applications classiques et présente une spécificité relative à la gestion d'indices sous forme d'historiques. Cette spécificité a un impact sur l'ensemble des tâches de conception du système : représentation, organisation de la mémoire et phases de raisonnement.

3.3 Démarche générale de l'étude

Notre démarche propose de dépasser les limites des approches existantes qui restent liées à des applications spécifiques au sein de la classe de problèmes visée (cf. chapitre II). Plus précisément, nous proposons de définir un modèle d'indexation *générique* ainsi qu'un outil logiciel *ouvert* facilitant son utilisation au sein de systèmes de RàPC.

3.3.1 Nécessité d'un modèle d'indexation générique

La gestion de situations comportementales a un impact majeur sur les solutions à employer pour le *problème de l'indexation des cas* dans un système RàPC. L'identification des indices effectifs, parmi le nombre important de données d'observation disponibles (indices potentiels), est rendue plus difficile. La comparaison des indices intégrant des *séquences de valeurs* est nécessaire pour la phase de recherche (indices de recherche) et pour la phase de réutilisation (indices d'adaptation). Durant l'apprentissage, la sélection des indices effectifs qui guideront les raisonnements futurs est déterminante et la représentation spécifique des indices doit être prise en compte. En définitive, la classe de problèmes abordée est plutôt caractérisée par un certain type de données, les historiques, que par un type de tâches (diagnostic, interprétation, planification) ou par un type de connaissances. La définition du formalisme de représentation des indices et de leur gestion, que nous appelons *modèle d'indexation*, est centrale dans le cadre de cette classe de problèmes. De plus, un modèle d'indexation est aussi un prérequis sur lequel les méthodes de raisonnement spécifiques ou génériques viendront s'appuyer.

Un exemple d'approche similaire est donné par le modèle d'indexation UIF (*Universal Index Frame*, Kolodner, 1993, pages 219–244). Ce modèle d'indexation aborde la classe des problèmes liant des agents et des buts à atteindre. Un ensemble d'attributs est alors défini pour indexer des expériences : l'émotion, la tâche, l'objectif de la personne engagée dans une situation. Ces descriptions générales peuvent alors être utilisées dans différentes applications issues de cette classe de problèmes. Plus généralement, il apparaît nécessaire « de pouvoir identifier des classes de problèmes et de créer un modèle d'indexation pour chacun » (Kolodner, 1993, page 244). Le modèle UIF aborde une classe de problèmes très différente de la nôtre, dans laquelle il n'y a pas de notion d'historique. Ce modèle n'aborde pas non plus le problème de l'indexation dans son ensemble comme nous l'avons défini (cf. §2.1.3). Cependant, cette approche partage avec la nôtre la poursuite de deux objectifs importants :

[17] Ces systèmes sont décrits et analysés dans (Kolodner, 1993).

- faciliter l'analyse de l'application du RàPC à un type de problèmes par la définition d'un formalisme générique,
- permettre le développement d'une compétence méthodologique accumulée à travers l'analyse de différentes applications.

3.3.2 Besoin d'un outil logiciel ouvert pour la gestion du modèle et du RàPC

Le besoin d'outils est reconnu dans la communauté scientifique pour faciliter la spécification, le développement et la mise en place de systèmes RàPC. Janet Kolodner s'exclame d'ailleurs dans la conclusion de son livre « Nous avons besoin d'outils ! » (Kolodner, 1993, page 571) et ce besoin a été plus récemment souligné (Leake, 1996, pages 17–18). Dans notre démarche, nous visons la conception d'un outil pour faciliter :

- l'application du modèle d'indexation proposé,
- la capitalisation de composants logiciels liés à notre modèle et plus largement au RàPC.

La définition d'un modèle d'indexation permet de construire un cadre adapté à une classe de problèmes. Cette démarche descendante permet de faciliter le développement d'applications particulières issues de cette classe de problèmes. Cependant, l'utilisation concrète de ce type de modèle doit prendre en compte les connaissances spécifiques et les techniques applicables dans le cadre d'une application donnée. Ainsi, il est possible d'optimiser certaines opérations ou d'utiliser des hypothèses particulières. L'outil gérant ce modèle doit donc permettre sa *spécialisation* (cf. Figure I-8). De plus, un modèle d'indexation n'aborde qu'une partie de la problématique de la construction d'une application de raisonnement à partir de cas. Il s'agit alors d'effectuer une *intégration* de ce modèle avec d'autres techniques et approches nécessaires pour réaliser le système complet (contrôle et exécution des phases de recherche, de réutilisation, de révision et d'apprentissage). Ainsi, durant les opérations de spécialisation et d'intégration, cet outil doit aider à la gestion de la cohérence et de l'implantation d'un système final. Enfin, cet outil ne doit pas se restreindre à la gestion du modèle d'indexation proprement dit, et doit également gérer l'application du RàPC de manière générale pour effectuer l'intégration.

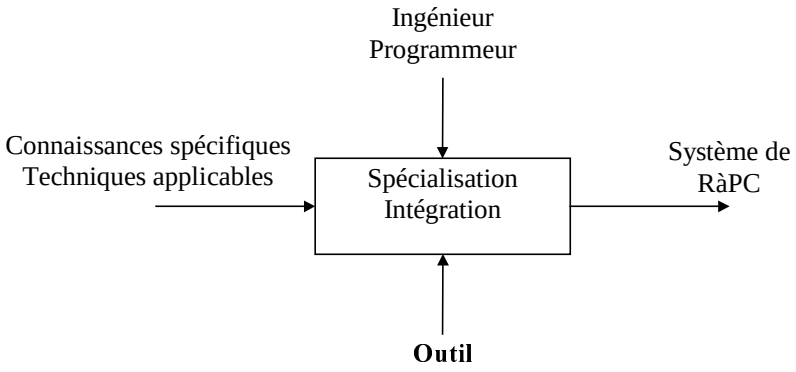

Figure I-8 : tâche de réalisation d'un système de RàPC[18]

Lors de ce processus de spécialisation et d'intégration, de nouveaux composants logiciels sont conçus : ils permettent d'étendre ou d'enrichir l'ensemble des composants existants. C'est pourquoi, nous voulons compléter cette démarche descendante par une *démarche ascendante* de capitalisation de composants. Au fur et à mesure de la réalisation d'applications, il s'agit de constituer un ensemble

[18] Ce diagramme utilise la notation SADT (Lissandre, 1990).

de composants réutilisables pour faciliter progressivement la gestion du modèle d'indexation et du RàPC.

Ainsi, un outil facilitant le processus de spécialisation et d'intégration ainsi que la capitalisation de composants, doit être réutilisable dans différents contextes d'applications et surtout *ouvert* : il doit en effet permettre l'extension des composants logiciels existants ainsi que l'ajout de nouveaux composants.