

## Des systèmes conseillers épiphytes

Sylvain Giroux\*    François Pachet\*\*    Gilbert Paquette\*    Jean Girard\*

\*LICEF, Télé-Université, 1001, rue Sherbrooke est, Montréal, Canada,  
sgiroux@teluq.quebec.ca

\*\*LAFORIA, Université Paris VI, 4place Jussieu, 75252 Paris Cedex 05, France,  
fdp@laforia.ibp.fr

### RESUME

Dans le cadre de la formation à distance, l'absence de tutorat direct exige la mise en place de ressources pour relancer un processus d'apprentissage en difficulté ainsi que des outils pour surveiller la progression des étudiants. De plus, un campus virtuel constitue l'exemple par excellence d'un environnement de formation réparti. A leur manière, les systèmes multi-fenêtres actuels sont aussi logiquement répartis. Dès lors il n'est pas surprenant qu'une même plate-forme EpiTalk serve pour le développement de systèmes conseillers dans les deux cas de figure. Cependant il faut choisir avec soin les éléments qui sous-tendent cette plate-forme. Dans EpiTalk, ce sont les agents, les organisations et les graphes hiérarchiques. Les agents consacrent l'autonomie de chaque composante du système conseiller; cette autonomie est essentielle dans les systèmes répartis. Les organisations lient les agents entre eux à l'exécution. Les graphes hiérarchiques structurent la description des systèmes conseillers et dictent à l'exécution les voies de communication entre agents. Ainsi EpiTalk sert tout autant la description des systèmes conseillers que leur génération et leur activation. Comme les systèmes conseillers sont greffés sur les applications existantes et évoluent de concert avec celles-ci, ils sont qualifiés d'*épiphytes* par analogie avec la botanique.

### 1. Introduction

Déjà dans le cadre de la formation à distance "traditionnelle", l'absence de tutorat direct exige la mise en place de ressources pour relancer un processus d'apprentissage en difficulté ainsi que des outils pour surveiller la progression des étudiants. Ces ressources peuvent être plus ou moins interventionnistes dépendamment de la liberté accordée à un étudiant. Elles s'adressent aussi bien à l'activité d'apprentissage comme telle, qu'à

l'utilisation des outils d'apprentissage. De surcroît, les moyens de communication actuels brisent l'isolement qui était le lot des étudiants d'hier, mais posent de nouveaux défis pour la formation à distance. Manifestement les autoroutes de l'information, le courrier électronique, la vidéoconférence, rendent désormais envisageable la collaboration à distance entre individus pour l'accomplissement d'une tâche. Toutefois cette délocalisation exige des outils de support à la tâche, que ce soit pour faciliter la navigation dans des univers informatiques de plus en plus complexes ou pour aider les individus à réaliser seul ou en groupe une tâche.

Les systèmes conseillers sont appelés à devenir un des éléments centraux de l'arsenal dont dispose la formation à distance. C'est pourquoi à la Télé-Université nous avons développé une architecture générique pour le développement de systèmes conseillers. Un système conseiller observe le travail d'un utilisateur, analyse ses actions et prodigue, sur demande de l'apprenant ou à sa propre initiative, des conseils de nature à améliorer la production ou la démarche de l'utilisateur. Ces conseillers doivent pouvoir être greffés autant sur des applications existantes que sur de nouvelles applications. Ces applications peuvent tout aussi bien être développées par Télé-Université, e.g. HyperGuide [Paquette, et al., 93a], que par des sociétés privées, e.g. Word de Microsoft. Dans le premier cas, le code source est disponible, ce qui facilite l'intégration des conseillers. Dans le second cas, le code source reste inaccessible. C'est dans ce cadre que nous avons développé EpiTalk, une plate-forme pour la description, la génération et l'activation de systèmes conseillers individuels ou collaboratifs.

Les systèmes conseillers constituent un cas particulier d'une catégorie plus vaste de systèmes d'information [Giroux et al., 94a]. La principale préoccupation de ces systèmes n'est plus l'interaction avec un usager, mais plutôt l'observation du comportement d'un système d'information existant conjuguée à la structuration et à l'analyse de ces observations. C'est le cas par exemple des dévermineurs [Giroux, et al. 94b]. Par analogie avec la botanique, nous qualifions d'*épiphytes* de tels systèmes d'information: ils *poussent* sur les autres systèmes d'information.

En résumé, le contexte spécifique de la formation à distance à Télé-Université nous a amené à considérer d'une part les systèmes conseillers dédiés à un seul apprenant et d'autre part, les systèmes conseillers répartis dédiés au travail collaboratif entre plusieurs apprenants et formateurs. De plus, nous désirions continuer à bénéficier des outils déjà développés. Les systèmes conseillers développés devaient pouvoir se greffer sur les applications déjà existantes. Ces desiderata ont guidé le design et le développement de

EpiTalk. EpiTalk fournit un cadre uniforme applicable autant pour la formation individualisée que pour la formation dans le cadre d'activités collaboratives.

Dans cet article, nous étudions d'une part les systèmes conseillers en tant que systèmes épiphytes et d'autre part la description et la génération de tels systèmes. EpiTalk y est présenté dans son ensemble, c'est-à-dire autant dans son état actuel que dans les développements en cours ou à venir. Dans un premier temps, nous distinguons les différentes composantes mises en oeuvre dans le cadre des systèmes épiphytes. Ensuite, nous montrons comment EpiTalk permet de décrire, de générer, de greffer et d'activer des systèmes conseillers épiphytes. Par après, nous examinons la fonction de conseil. Finalement, nous esquissons les conseillers collaboratifs répartis.

## **2. Les systèmes conseillers en tant que systèmes épiphytes**

Dans les environnements informatisés d'apprentissage, on distingue différents styles d'intervention dans le processus d'apprentissage [Wenger 87] . A une extrémité se trouvent les systèmes qui ne donnent qu'une simple *rétroaction* à l'apprenant. Un niveau interventionniste légèrement plus élevé consiste à prévoir une aide interactive intelligente, e.g. afficher sur demande de l'utilisateur une explication relative à la composante de l'interface où il se trouve ou à l'outil qu'il utilise. Un niveau interventionniste encore plus élevé correspond aux systèmes de type *coach* ou *les conseillers actifs*, e.g. [Brown, et al., 82] [Shute et Glaser, 90]. Le système possède alors des connaissances sur la tâche ou le problème à résoudre. Non seulement l'apprenant peut-il demander une aide ciblée sur ses activités, mais le système peut également intervenir pour afficher un conseil lorsqu'il lui semble que l'apprenant éprouve des difficultés trop grandes avec la tâche en cours. Cependant, le conseil n'est pas impératif: il peut être suivi ou non par l'apprenant. Le conseil peut porter sur le contenu spécifique d'un domaine ou sur une méthodologie liée à une tâche générique. Enfin, dans les systèmes tutoriels intelligents, l'initiative entière est laissée au système qui exerce un tutorat serré. Les interventions du système sont impératives, les erreurs sont soulignées et exigent correction.

Dans le design de systèmes conseillers, il s'agit de trouver le bon dosage d'intervention du système en fonction des besoins de formation des apprenants. Plus le conseiller est actif ou carrément interventionniste, moins l'apprenant a de marges de manoeuvre pour exercer son activité cognitive. A l'inverse, plus l'apprenant est actif, plus le conseiller est passif, moins il intervient. Trop de conseils et l'apprenant n'a plus d'espace pour

apprendre; trop peu de conseil et la plupart des apprenants piétineront dans des voies peu productives.

Le design de systèmes conseillers a donné lieu à de nombreuses recherches, e.g. [Wilson et Jonassen 90] [Paquette 92]. Cependant la plupart des systèmes conseillers ont été développés soit comme des systèmes *self-contained*, soit comme composantes d'environnements pédagogiques intégrés. Le design du système conseiller est alors enseveli sous le design de grands systèmes pédagogiques, ce qui rend difficile leur transposition à d'autres contextes. Dans le cadre de notre approche, nous considérons les systèmes conseillers comme des composantes indépendantes et non plus comme des parties d'un système. En fait, les systèmes conseillers sont vus comme des prolongements d'applications existantes, qui, à l'exemple d'un traitement de textes, n'ont pas comme but premier de faire partie d'environnements de formation. Sous EpiTalk, un système conseiller devient un système d'information qui observe le comportement d'un autre système d'information et qui réfléchit à partir de ses observations.

Les plantes épiphytes sont des plantes qui croissent sur d'autres végétaux sans leur causer de préjudices, à l'opposé des plantes parasites. Aussi par analogie, nous qualifions d'*épiphytes* ces conseillers. Cet adjectif, *épiphyte*, résume bien la relation qui unit le système d'information observé -l'application hôte- et le système d'information observant -le conseiller-, soit

- le conseiller ne peut exister sans application hôte;
- l'application hôte peut exister sans conseiller ;
- l'application hôte et le conseiller ont des existences indépendantes;
- le conseiller ne porte pas préjudice à son hôte.

En outre, les plantes épiphytes s'organisent souvent en véritables écosystèmes. Cette organisation en écosystèmes de conseillers permet la coexistence de points de vue multiples sur une même situation, la structure de chaque conseiller exprimant alors un point de vue particulier et partant, la communication d'informations entre conseillers traduit les échanges entre points de vue différents sur une même problématique.

Un conseiller épiphyte, est donc par métaphore botanique, un système d'information qui se greffe sans affecter son hôte. Le conseiller possède sa propre structure. C'est un

système d'information qui raisonne et agit à partir des interactions observées entre les composantes de l'application hôte.

### **3. De la nature multi-agents des systèmes conseillers épiphytes**

Conseillers et hôtes possèdent donc des propriétés d'indépendance les uns par rapport aux autres. Cette indépendance est à comprendre dans deux sens :

- au sens conceptuel :

l'architecture du conseiller est a priori indépendante de celle de l'hôte. Leurs architectures ne sont aucunement semblables ou isomorphes. En particulier, la structure de l'hôte ne ressemble en rien à la structure des conseillers greffés.

- au sens du génie logiciel :

les hôtes ne sont aucunement conçus en fonction d'un conseiller particulier. Cette contrainte est importante, car elle reste le gage de la réutilisabilité des hôtes en tout ou en partie. Tout hôte programmé en fonction d'un conseiller précis perd ses propriétés de réutilisabilité.

Cette propriété d'indépendance amène donc à considérer hôte et conseiller comme autonomes. Ceci constitue un premier argument en faveur des systèmes multi-agents comme élément de base des conseillers.

Egalement dans le cadre des environnements de formation, maints facteurs conduisent tout aussi naturellement vers des architectures multi-agents:

- la multiplicité des niveaux d'expertise;
- la distribution physique de certains hôtes, e.g. les "campus virtuels"<sup>1</sup> [Hiltz 86, 90] [Paquette, et al., 93a];
- la distribution logique des hôtes, e.g. les systèmes multi-fenêtres;

---

<sup>1</sup> Contrairement aux campus traditionnels où les étudiants sont regroupés physiquement dans un même lieu, un campus virtuel les regroupe virtuellement, chacun participant à la vie scolaire à partir de sa station de travail située dans son domicile. La formation à distance telle que pratiquée à Télé-Université procède de la même problématique. La formation à distance est particulièrement cruciale dans les pays et les régions à faible densité de population.

- la nature même du processus humain de résolution de problèmes; même si les preuves sont présentées linéairement, leur découverte repose sur de nombreux allers-retours et s'éloigne nettement d'un processus linéaire.

Finalement, en "déglobalisant" l'expertise et les attributions, les systèmes multi-agents facilitent la tâche des concepteurs de systèmes conseillers en proposant un cadre plus approprié autant au niveau conceptuel qu'au niveau matériel. Chaque agent épiphyte<sup>2</sup> est chargé d'une tâche spécifique dont la granularité dépend de sa position dans une hiérarchie. Les agents au sommet de cette hiérarchie possède un point de vue global sur une situation alors que les agents au bas de cette hiérarchie voient leur champ de perception restreint. Ainsi certains agents sont chargés d'une tâche bien délimitée (e.g. surveiller l'activité d'un seul agent du système hôte), et d'autres voient leurs activités englober un ensemble de tâches ou d'actions (e.g. s'assurer que la démarche d'un apprenant corresponde à la démarche scientifique).

#### 4. EpiTalk

EpiTalk permet de spécifier, de générer et de surveiller l'évolution des systèmes conseillers pour toute application développée en Smalltalk-80. EpiTalk est implémenté en Smalltalk-80. Il constitue un prolongement de Actalk pour sa partie acteurs [Briot, 89], de ReActalk pour sa partie agents et réflexivité [Giroux, 93] et de NéOpus [Pachet, 92] pour sa partie représentation des connaissances et raisonnement. Un conseiller est greffé sur le système hôte à observer par réflexivité. Les conseillers sont générés automatiquement à partir d'une description du système à observer.

EpiTalk constitue une architecture opérationnelle, comportant tous les maillons de la chaîne, de la conception abstraite du conseiller à son implémentation. Cette architecture propose en effet à la fois un *formalisme de représentation* et un *vocabulaire* (basé sur les graphes hiérarchiques), ainsi qu'une *mécanique* d'interprétation de ces graphes. A l'instar de la plupart des systèmes de spécification, EpiTalk distingue entre la phase de spécification et la phase d'exécution pendant laquelle l'hôte est observé. Le système hôte est observé par espionnage. Finalement EpiTalk supporte la description, la génération et l'activation des conseillers à travers une pléiade d'outils et d'interfaces de développement.

---

<sup>2</sup> Comme un conseiller épiphyte est un système multi-agents, les agents qui le composent sont dits épiphytes.

Les descriptions dans EpiTalk sont à base de graphes; d'une part, les pédagogues s'avèrent familiers avec de tels modèles et d'autre part, maints travaux de modélisation en intelligence artificielle utilisent les graphes, e.g. les graphes conceptuels [Sowa 84]. A l'exécution, les systèmes conseillers sont des systèmes multi-agents. Ceux-ci autorisent un fonctionnement autonome de chaque conseiller ainsi que de chacune de ses parties. Ainsi la structure des systèmes conseillers peut évoluer de concert avec l'évolution de leur hôte: les conseillers épiphytes sont générés ou supprimés avec le lancement ou la fermeture des applications hôtes; les agents épiphytes qui appartiennent à un conseiller sont eux aussi générés ou supprimés avec la création et la suppression des agents hôtes. Les conseillers peuvent aussi s'échanger de l'information. Ainsi l'autonomie et la plasticité des systèmes multi-agents font en sorte que les systèmes conseillers s'accommoderont bien des architectures réparties envisagées.

EpiTalk repose sur la manipulation de trois graphes qu'il est important de bien distinguer afin de comprendre la suite de sa description. Il s'agit de la réification de l'hôte sous forme de graphe organisationnel (§5), du graphe des tâches qui décrit abstraitement un conseiller (§6) et du graphe des agents épiphytes qui représente un conseiller épiphyte pendant son activation (§7).

## **5. Le système hôte**

Du point de vue d'un système conseiller, un hôte n'existe qu'à travers les interactions qui s'y déroulent. En effet pour tout apprenant, seules sont observables les interactions qu'il a avec ses instruments de travail. Il n'est jamais possible de déterminer avec certitude ses connaissances et ses processus de raisonnement. Le design de EpiTalk et des conseillers générés procède de ce point de vue: dans le cadre d'une application Smalltalk-80, un conseiller ne pourra raisonner qu'à partir des messages qui transitent entre les objets qui composent un hôte. Pour cela, il faut d'une part réifier les messages qui transitent à l'intérieur du système hôte et d'autre part spécifier les conseillers et leurs actions à partir de ces seules observations. Cette section traite de la réification des hôtes. La section 6 traite de la spécification des conseillers à partir des informations réifiées (§7.3).

### **5.1 La réflexivité**

La réflexivité permet de raisonner sur soi afin d'agir sur son propre comportement. Elle s'applique soit introspectivement sur soi-même et sur son processus interne de pensée, soit extérieurement sur son comportement et sa situation dans le monde [Smith, 1983].

Lors d'un dessalage en canot, afin de corriger la fausse manoeuvre et apprendre de cette expérience malheureuse, le canotier a besoin d'une représentation des actions liées au dessalage. De manière analogue, pour apprendre de ses erreurs, un programme doit avoir accès à son contexte. Par exemple, en 3-Lisp [Smith, 1983], ce contexte est composé du texte de l'expression, de l'environnement et de la continuation.

La clé de voûte de la réflexivité réside dans la correspondance qui s'établit entre le niveau dénoté et le métaniveau: le métaniveau constitue une description réifiante, manipulable et auto-référentielle du niveau dénoté. La réification explicite certains aspects cachés ou inaccessibles au niveau inférieur. De plus, il existe un *lien causal* entre le niveau dénoté et le métaniveau; toute modification de la dénotation est reflétée au métaniveau et toute modification du métaniveau est reflétée au niveau dénoté. Finalement les univers réflexifs sont des univers en couches. Du point de vue d'un niveau, le niveau immédiatement inférieur est appelé la *dénotation*. Inversement du point de vue d'un niveau, le niveau immédiatement supérieur est qualifié de *méta*. Tout l'art et tout l'intérêt de la réflexivité réside dans le choix des aspects qui sont explicités.

Dans le cas qui nous préoccupe, l'activité de conseil peut être perçue dans une certaine mesure comme une activité réflexive. Nous retiendrons de la réflexivité principalement la réification du système hôte et des interactions qui s'y déroulent. Un conseiller épiphyte exercera donc ses activités à partir de la description réflexive de son hôte.

## 5.2 La réification des systèmes hôtes

Une application correspond à un système hôte. Chaque lancement d'une application donne lieu à l'instanciation et à l'activation des différents objets qui la compose. Le lancement d'une application peut être vu comme une instanciation de l'application. Par conséquent, une application donnée est décrite par une classe dans EpiTalk. De même, la structure de l'application est réifiée au niveau de la métaclasse. La métaclasse contient par exemple les classes des objets composant l'application et les différents points d'entrée accessibles par un usager pour lancer l'application. Au niveau d'une activation particulière, les objets en cause sont réifiés sous forme d'organisation sous ReActalk.

Dans EpiTalk, ces informations réflexives, autant structurelles qu'organisationnelles, sont utilisées pour la construction d'un "éditeur de l'application" qui permet d'une part de lancer l'application à travers ses divers points d'entrée et d'autre part, de visionner l'état opératoire d'une application, i.e. quels objets sont actifs, quels sont les chemins de communication entre ces objets (Fig. 1). La structure de l'hôte est réifiée par un graphe.



Ce graphe n'existe qu'à l'exécution et évolue avec l'hôte. Les nœuds correspondent aux agents hôtes<sup>3</sup> existants. Chaque fois qu'un agent hôte est créé, un nœud correspondant est ajouté. Les arcs représentent les réseaux de communication.

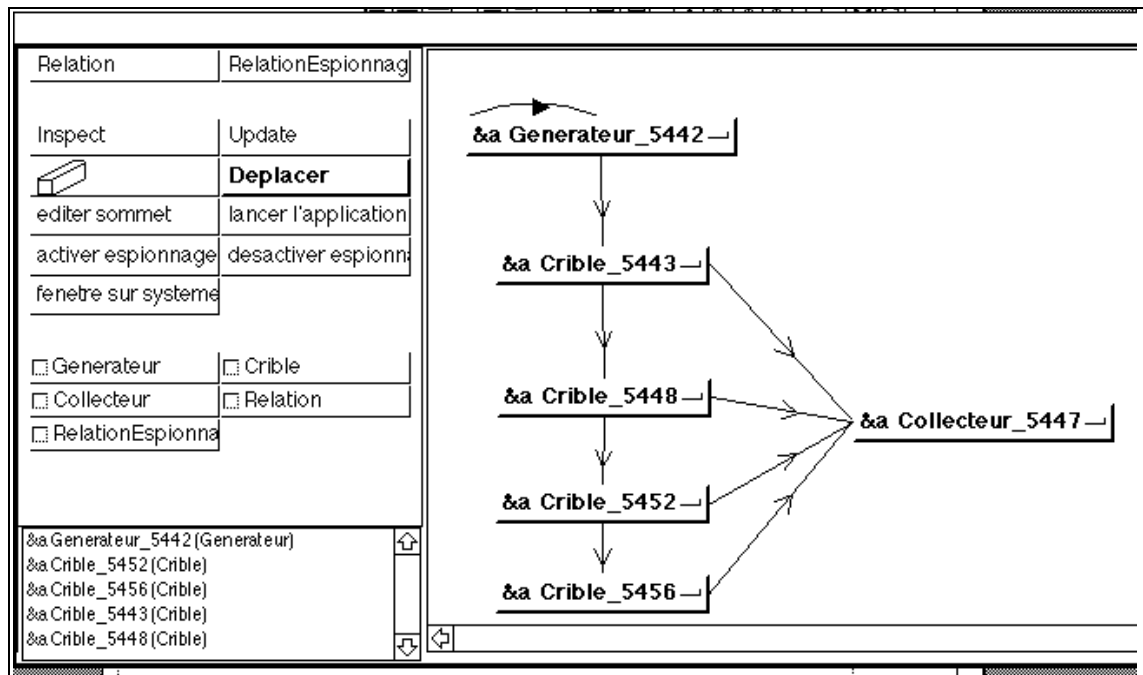


Fig. 1 Un éditeur sur une activation de l'application "Crible d'Erathostène" [Abelson et Sussman, 85].

### 5.3 Le lien causal entre un hôte et sa réification

En réflexivité, le lien causal assure la correspondance entre la dénotation et sa description réflexive au métaniveau. Il faut recueillir et garder à jour les informations réflexives présentées par l'éditeur. Cela peut parfois poser certains problèmes techniques, si on tient à ne pas modifier les applications afin que conseillers et hôtes conservent leur indépendance.

Dans le cas où une application donnée a été développée dans le cadre proposé par ReActalk (Fig. 1), cette gestion est aisée car le métaniveau est automatiquement informé par les agents de l'application de toute création d'agents et de tout nouveau lien de communication.

Le cas où l'application a été développée dans un cadre indépendant de ReActalk, e.g. l'environnement de programmation de Smalltalk-80 (Fig. 2) est évidemment beaucoup plus délicat à traiter, puisque nous nous sommes interdits toute modification d'une

<sup>3</sup> Nous appelons "agents hôtes" les objets qui composent un hôte.

application existante. Cette contrainte est importante à respecter si l'on veut que la méthodologie et les outils développés restent valides pour les applications auxquelles nous n'avons pas accès au code source. Pour suivre l'évolution d'une application, nous disposons par contre de son point d'entrée via l'éditeur. Aussi tout objet créé est-il immédiatement espionné [Pachet et al. 95]. Les espions réifient les messages qui transitent dans l'application. Ces messages réifiés tiennent lieu d'observation du comportement de l'utilisateur. Cet espionnage systématique permet de repérer les nouveaux agents hôtes<sup>4</sup>, de suivre l'établissement des liens de communication<sup>5</sup> et partant de suivre l'évolution de l'hôte et d'observer les interactions qui y prennent place.

Jusqu'à présent, nous avons concentré nos efforts sur les applications développées en Smalltalk-80. Cependant il nous apparaît légitime d'espérer que cette approche puisse être prolongée vers la construction de conseillers greffés sur des applications commerciales. En effet, les systèmes d'exploitation actuels commencent à donner accès à des informations de nature similaire à celles qui sont espionnées, e.g. les AppleEvents dans le cas du MacIntosh.

---

<sup>4</sup> Pour repérer les nouveaux agents hôtes, ceux-ci doivent par contre être rendus en résultat des transmissions de messages qui ont provoqués leur création.

<sup>5</sup> Au niveau de l'implantation, il suffit de connaître les objets faisant partie de l'hôte; les réseaux de communication peuvent être déduits à l'aide des messages espionnés et de facilités réflexives [Foote et Johnson, 89] de Smalltalk-80. Cependant pour l'instant, les liens entre les noeuds n'interviennent pas dans la construction et le raisonnement d'un conseiller.

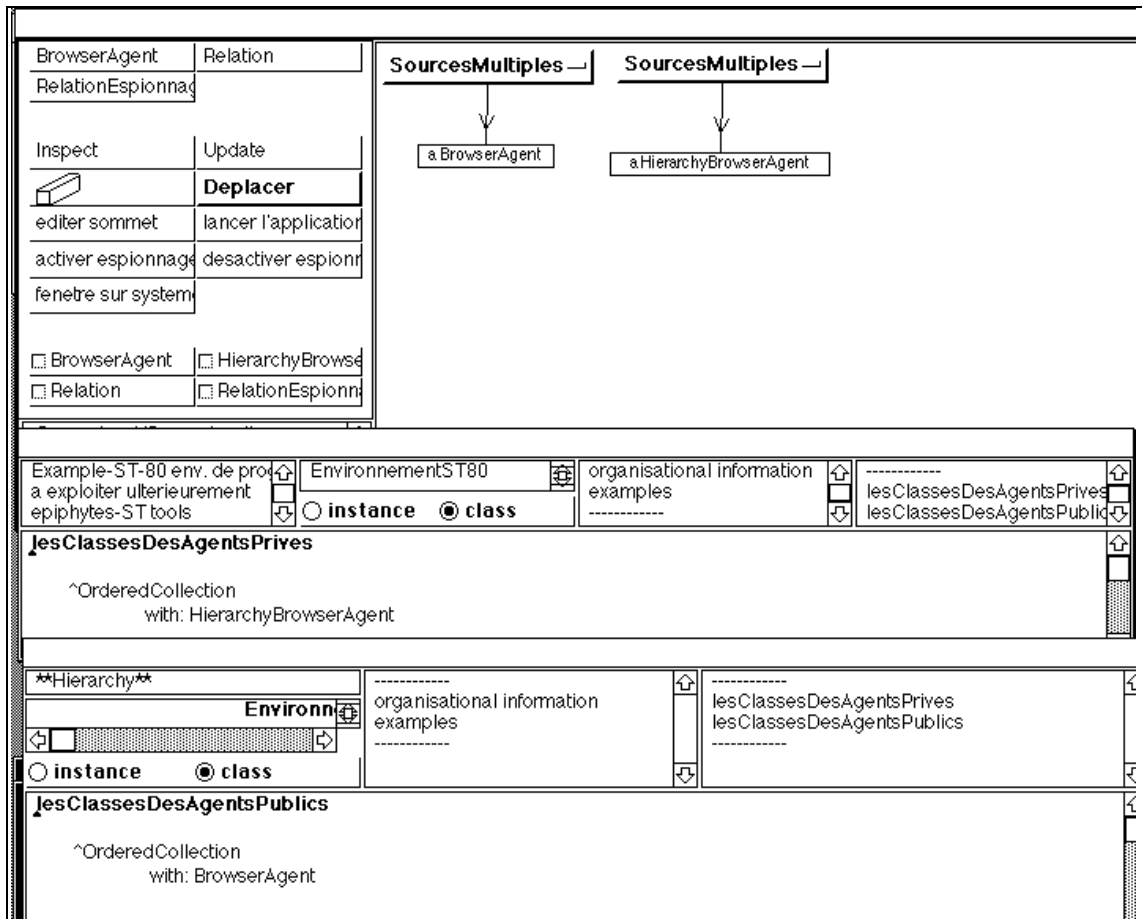


Fig. 2 Editeur d'application sur l'environnement de programmation de Smalltalk-80.

## 6. La description du système conseiller

Un système conseiller observe le travail d'un usager, analyse ses actions et prodigue, sur demande de l'usager ou à sa propre initiative, des conseils de nature à améliorer la production ou la démarche de l'usager. Dans EpiTalk, les graphes agissent comme formalismes fédérateurs pour exprimer différents points de vue sur un système hôte: son organisation (§5), l'objet qui est produit par l'utilisation de cette application et le processus qui a mené à cette production. Dans cette section, nous montrons comment un système conseiller est décrit à l'aide d'un *graphe des tâches*. Ce graphe des tâches peut être aussi bien utilisé pour décrire un conseiller portant sur l'objet produit par l'usager, e.g. la cohérence d'un cours dans le cadre d'un atelier de génie didacticiel [Paquette, et al., 94], que pour décrire un conseiller analysant la démarche d'un apprenant, e.g. la démarche d'un apprenant dans le cadre de son utilisation d'une application dédiée à l'apprentissage de la chimie [Paquette et al., 93b]. Le graphe des tâches présidera par la suite à la génération du système conseiller épiphyte.

## 6.1 La représentation de l'expertise sur la tâche

La description d'un conseiller est centrée sur un *graphe des tâches* (Fig. 3). Ce graphe sert à spécifier l'architecture du conseiller de manière abstraite. Tout conseiller est généré à partir de ce graphe:

**le graphe des agents épiphytes est isomorphe au graphe des tâches.**

Plus précisément, cet isomorphisme a deux conséquences :

- A chaque tâche du graphe des tâches est associé, à l'exécution, un agent épiphyte et un seul.
- La structure du graphe des agents épiphytes est la même que celle du graphe des tâches, i.e. les agents épiphytes auront la même structure hiérarchique que les tâches qu'ils surveillent.

D'une part, le graphe des tâches explicite le point de vue qu'a le concepteur de l'hôte et, d'autre part, ce point de vue détermine le processus d'analyse, réalisé par les agents épiphytes, des interactions se produisant à l'intérieur de l'hôte.

Un graphe des tâches possède comme caractéristiques :

- Le graphe est hiérarchique. Il n'y a pas de cycle.
- Les "racines" du graphe sont les nœuds sans supérieur hiérarchique.
- La relation père-fils est une relation de hiérarchie de parties : un nœud non terminal "contient" tous ses fils, et les descendants de ses fils, récursivement.

Nous distinguons trois sortes de nœuds dans un graphe des tâches:

1- les nœuds "terminaux", ou les feuilles.

Ces nœuds sans fils possèdent un statut particulier car les agents de l'hôte sont associés à ces nœuds. Seuls les nœuds terminaux du graphe des tâches se voient associer un ou plusieurs agents hôtes. Par convention, on représente les nœuds terminaux entre crochets, [].

2- Les nœuds non terminaux

Ce sont les nœuds qui ne sont pas des feuilles. On distingue encore deux possibilités :

2a- les nœuds non terminaux simples

2b- les nœuds "étoile".

Les nœuds "étoile" représentent des hiérarchies de partie dont le nombre de composants (de 0 à n) est inconnu a priori. A chaque création d'un agent hôte concernant la tâche, le sous-arbre correspondant est alors ré-instancié. Par convention, ces nœuds sont suffixés d'une étoile "\*".

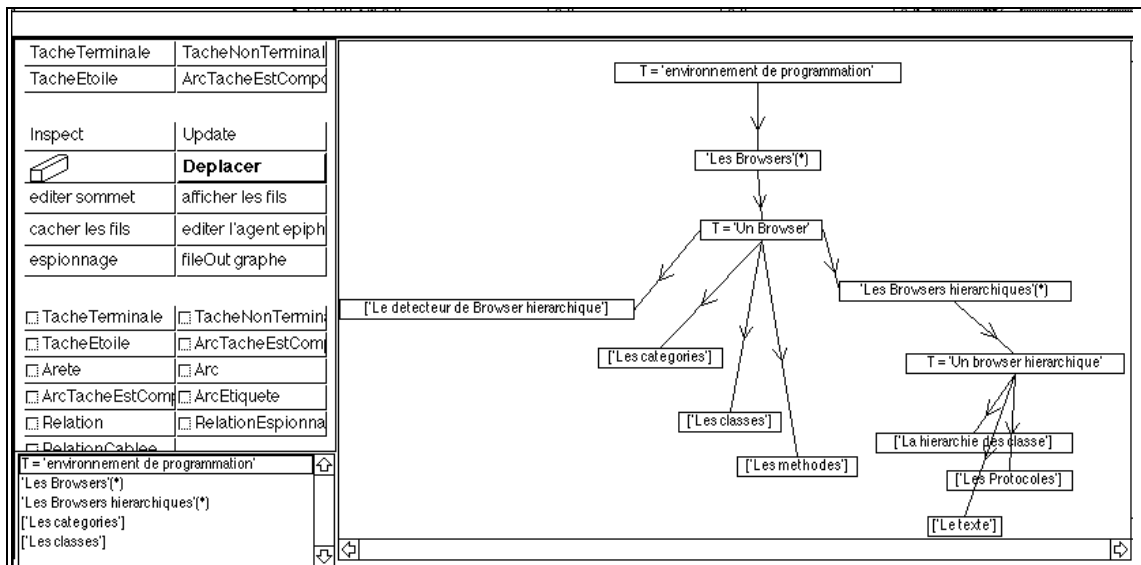


Fig. 3 Le graphe des tâches attaché à l'hôte "Environnement de programmation Smalltalk"

Le terme "graphe des tâches" est volontairement général pour embrasser toutes les descriptions qu'un programmeur désire énoncer sur un hôte existant. Cette description peut être une description des *tâches* à effectuer par un utilisateur ou par l'hôte<sup>6</sup> ou bien une description hiérarchique des *résultats* à produire. La sémantique d'un graphe des tâches est donc entièrement déterminée par le concepteur.

La mécanique d'instanciation des agents épiphytes est intimement liée au graphe des tâches et à sa structure. En effet, ce graphe guide la génération d'un conseiller au moment du lancement d'un hôte et au fur et à mesure de son évolution. Plus précisément, le graphe des tâches remplit deux fonctions :

<sup>6</sup> Par exemple, les tâches au sens de Chandrasekaran [Chandrasekaran 87].

- organiser à l'exécution les agents épiphytes dans une hiérarchie;
- pour chaque agent épiphyte, spécifier les agents hôtes à surveiller et, pour chaque agent hôte, spécifier les messages à capter (Fig. 8).

Ainsi EpiTalk n'exploite actuellement dans un graphe des tâches qu'un seul type de lien, celui de hiérarchie de parties, ou d'agrégation.

Pour conseiller, en plus de disposer de modèles de l'objet *idéal* sur lequel portent les conseils, il faut des modèles de la situation passée, immédiate et future. Ces modèles se traduisent pour l'instant dans la spécificité des opérations effectuées par les agents épiphytes. Ils sont donc décrits dans les comportements des agents épiphytes<sup>7</sup>. Bien sûr, ces modèles peuvent aussi être spécifiés via l'introduction de d'autres types de liens dans un graphe des tâches, comme des liens de précédence temporelle (telle tâche doit être effectuée avant telle autre), ou des liens conditionnels (Fig. 5). A travers de tels arcs, le passage d'un noeud à un autre situe les interactions dans le temps et sert à anticiper les actions à venir. Cependant pour l'instant, une contrainte de précédence entre deux tâches  $t_1$  et  $t_2$  s'exprime dans la tâche qui contient à la fois  $t_1$  et  $t_2$ . Le comportement de l'agent épiphyte associé à cette tâche spécifie leur ordonnancement correct ainsi que les actions à prendre en cas de violation. Actuellement les arcs du graphe des tâches ne sont pas exploités dans le raisonnement des agents épiphytes.

Bien évidemment, il existe autant de graphes des tâches pour un hôte qu'il existe de points de vue sur cet hôte. Comme leur architecture multi-agents permet aux conseillers d'opérer en parallèle, il est donc possible d'analyser en parallèle un hôte sous plusieurs points de vue.

## 6.2 La modélisation de la production

EpiTalk a été utilisé pour la description et la génération d'un système conseiller greffé sur un logiciel dédié au conception de cours, l'AGD: les conseils portent sur la cohérence des objectifs dans un cours en phase de design [Paquette, et al., 94]. Un cours possède un certain nombre d'objectifs et un certain nombre d'unités de formation. Une unité de formation possède un certain nombre d'objectifs et un certain nombre d'activité de formation, etc. (Fig. 4). Les conseils dispensés reposent sur cette structure hiérarchique. En voici quelque exemples:

---

<sup>7</sup> Loin d'être monolithique, un agent épiphyte peut être le résultat de l'assemblage de plusieurs sous-agents. Dans ReActalk, il existe un éditeur qui permet de spécifier et d'assembler les agents épiphytes à partir de blocs déjà définis, e.g. mémoire, analyseur de trace, base de règles.

Condition de déclenchement:

la compétence requise n'est pas déterminée et le niveau d'habileté est supérieur à 3

Conseil:

Il est toujours souhaitable de préciser les exigences qui entourent un objectif. Quand l'objectif est de niveau <n>, on exprime souvent ces exigences de façon quantitative. Par exemple, on précise une limite de temps, un nombre de réussites, un degré de précision ou un pourcentage d'erreur.

Condition de déclenchement:

plus de 30% des niveaux taxinomiques de tous les objectifs d'une même unité de formation sont inférieur ou égaux à 3 et le public-cible est de niveau au moins avancé

Conseil:

L'analyse des objectifs de cette unité de formation montre que <x%> des objectifs sont de niveau taxinomique élémentaire (niveau: mémoriser, comprendre ou appliquer). Compte tenu du domaine des connaissances et du niveau de votre public-cible, nous vous suggérons de cibler vos objectifs à un niveau plus élevé afin de répondre aux besoins de formation.

Le premier conseil est défini au niveau de la tâche ['Un objectif du cours'] (Fig. 4). Il repose sur un point de vue très local et est donc situé au niveau d'une tâche terminale. Le second conseil est défini au niveau de la tâche 'Les objectifs d'une unité de formation' (\*) (Fig. 4). Il repose sur un point de vue plus global et se situe plus haut dans la hiérarchie..

Au niveau de la fonction de conseil, l'intervention la plus pertinente est déterminée en fonction de la proximité entre le dernier objet modifié et le ou les objets touchés par le conseil. Les liens entre les objets à analyser sont déterminés par des arcs de dépendance.

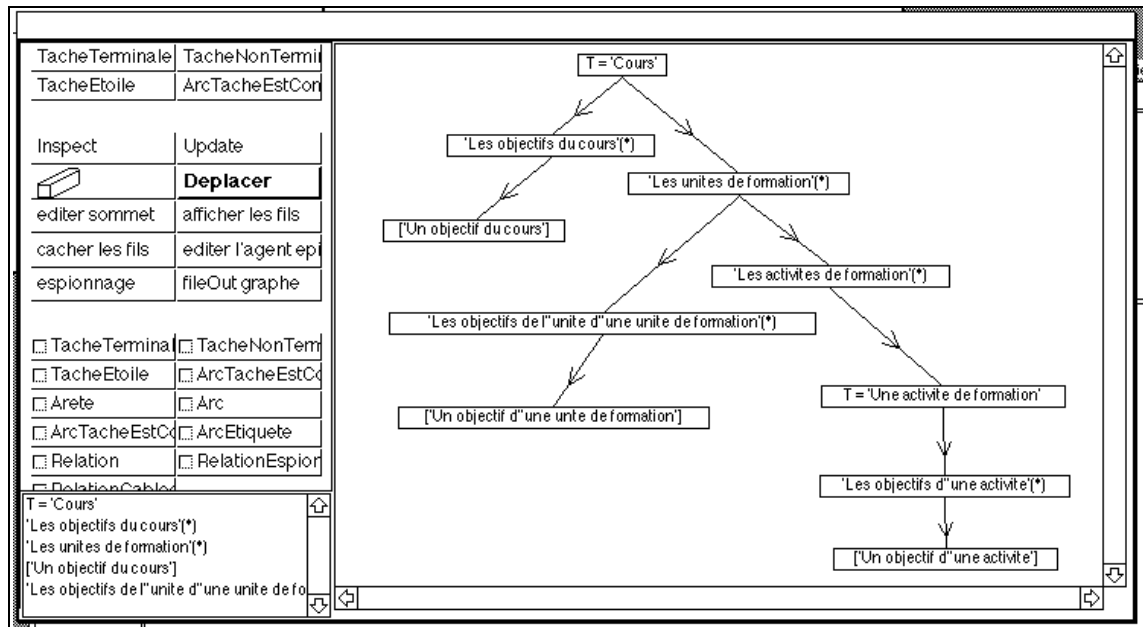


Fig. 4 Le graphe des tâches attaché à la modélisation d'une production.

### 6.3 La modélisation de la démarche

EpiTalk est utilisé pour la description et la génération d'un système conseiller greffé sur un environnement d'apprentissage pour l'expérimentation en chimie: le système conseiller observe la démarche de l'apprenant et vérifie s'il se conforme à la démarche scientifique [Paquette et al., 93b]. Le graphe des tâches de la Figure 5 indique les relations de précedence qui sont utilisées par le conseiller pour générer ses conseils. L'intervention la plus pertinente est déterminée en fonction des actions anticipées. Les séquences d'actions sont déterminées à la conception par ces arcs de précedence. Voici des exemples abrégés de conseils dispensés:

- #1 La loi que vous proposez n'a pas été validée; avant de proposer une loi, vous devriez essayer de produire et analyser plus d'observations.
- #2 Votre analyse n'est pas complète puisque vous n'avez effectué aucun regroupement d'observation à travers un graphe ou un tableau.
- #3 Votre sélection va créer un regroupement d'observations qui sera difficile à analyser parce que vous avez trop de variables.



Le conseil #1 repose sur l'ordonnancement des tâches. En effet, un arc *précède* relie la tâche 'Validation de la conjecture' à la tâche 'Généralisation' (Fig. 5). Le conseil #2 repose sur la décomposition de la tâche 'Analyse des observations' en sous-tâches. Cette décomposition n'est pas présentée dans la Figure 5. Cette décomposition permet de tracer au niveau d'une tâche les sous-tâches qui ont été ou n'ont pas été réalisées. Le conseil #3 repose sur des informations locales. La tâche correspondante est une sous-tâche terminale (non présentée dans la figure) de la tâche 'Analyse des observations'.

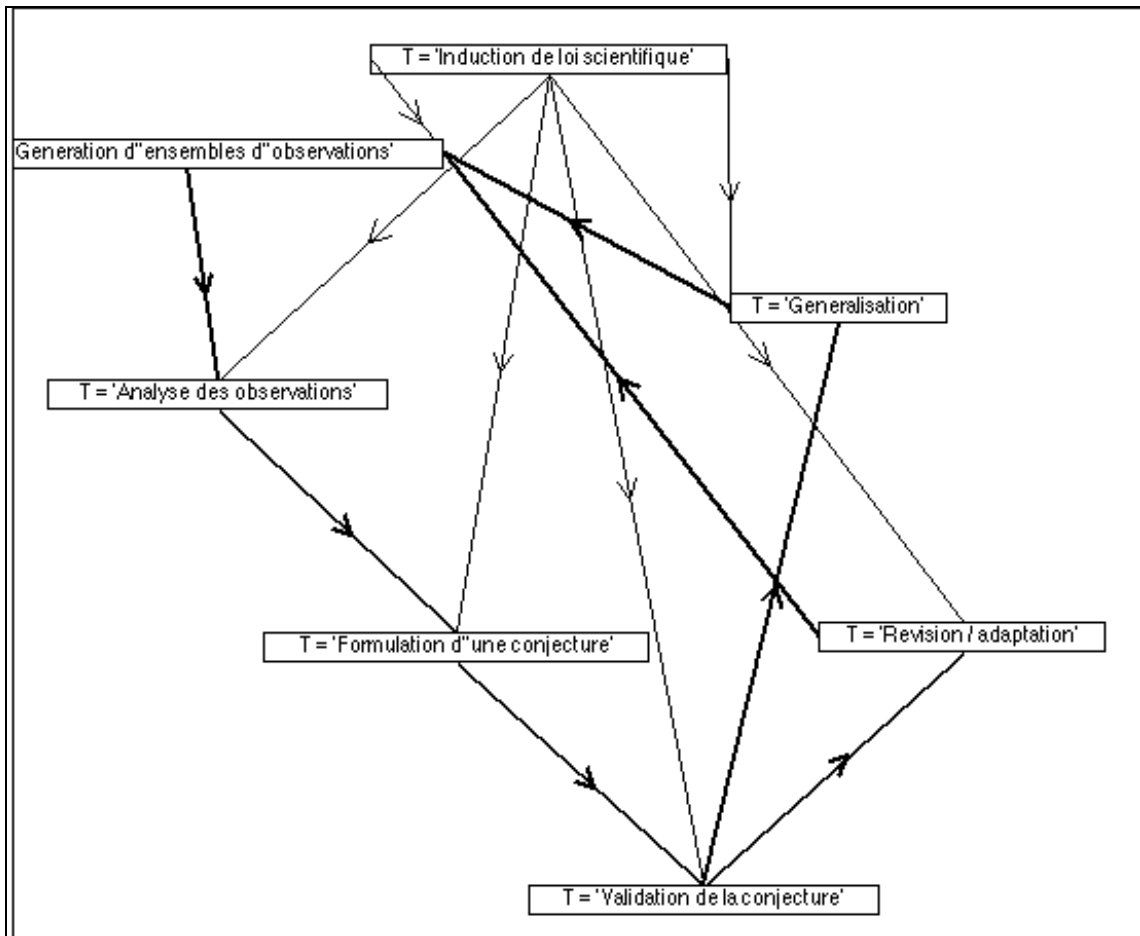


Fig. 5 Le graphe des tâches attaché à la modélisation de la démarche scientifique. Un seul niveau de décomposition des tâches est présenté. La relation de décomposition est décrite par les arcs fins. La relation de précédence est décrite par les arcs épais.

#### 6.4 La coexistence de points de vue multiples

EpiTalk est utilisé pour la description et la génération d'un système conseiller greffé sur un système tuteur intelligent pour la géométrie [Leman, et al., 94] [Marcenac, et al. 94].

Le conseiller observe l'apprenant sous plusieurs points de vue. Chaque point de vue représente un raisonnement expert différent utilisé pour résoudre le problème. Chaque point de vue via le graphe des tâches associé donne naissance à un système conseiller autonome. Expert et apprenant sont comparés à partir de modèles par recouvrement. La gestion des conflits entre conseillers reste inexplorée.

## **7. Les systèmes conseillers épiphytes à l'oeuvre**

A partir du moment où les conseillers épiphytes ont été décrits à travers les graphes des tâches, il reste à les activer lors du lancement du système hôte auquel ils sont destinés. Pendant la vie d'un système conseiller épiphyte, EpiTalk supplée des interfaces pour observer le système hôte (Fig. 1) et les systèmes conseillers épiphytes (§7.1). Il gère aussi la génération du système conseiller (§7.2) ainsi que sa greffe sur son système hôte et l'espionnage (§7.3)

### **7.1 Le graphe des agents épiphytes**

Le dernier des trois graphes manipulés par EpiTalk est le graphe des agents épiphytes. Il n'existe de toute évidence qu'à l'exécution. Les agents de ce graphe sont générés automatiquement par EpiTalk à partir du graphe des tâches (Fig. 3) et en fonction des agents de l'hôte (Fig. 6-7). Conformément à l'hypothèse fondamentale de EpiTalk, la structure du graphe des agents épiphytes est isomorphe à celle du graphe des tâches. Cependant les graphes n'en demeurent pas moins tout à fait différents et distincts, le premier entrant en jeu pendant l'exécution, le second, pendant la phase de description.

Cet isomorphisme est réalisé simplement : chaque tâche (nœud du graphe des tâches) est associée à un (et un seul) agent épiphyte. Si le nœud est terminal, l'agent épiphyte est terminal. Si le nœud est non terminal, l'agent épiphyte correspondant est non terminal. De plus, les liens de hiérarchie du graphe des tâches sont transposés dans le graphe des agents épiphytes.

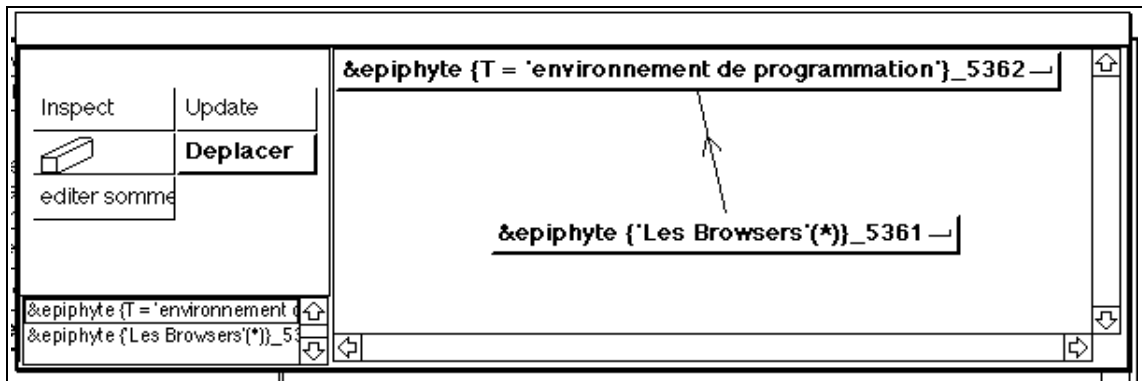


Fig. 6 Le système conseiller épiphyte correspondant à l'hôte "environnement de programmation de Smalltalk" à son lancement.

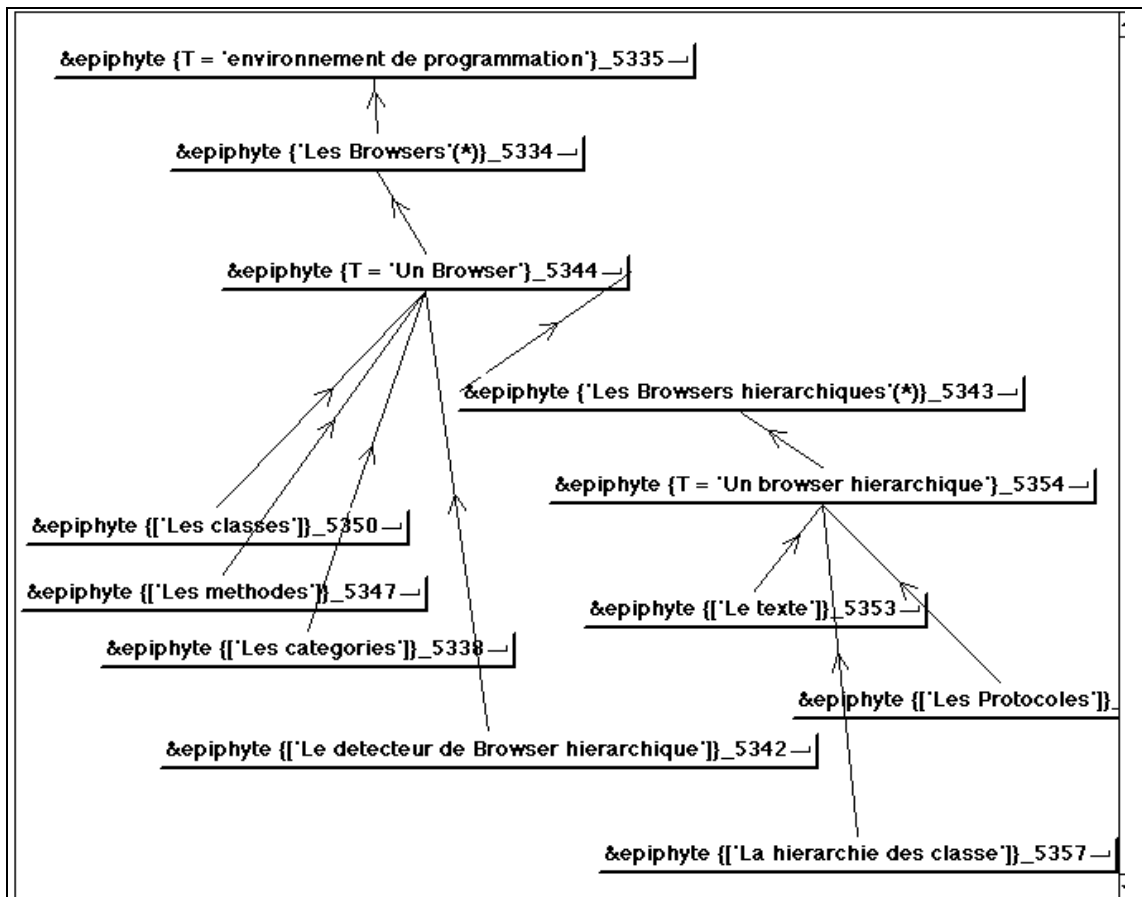


Fig. 7 Le système conseiller épiphyte correspondant à l'état de l'environnement de programmation de Smalltalk-80 de la Figure 2 et au graphe des tâches de la Figure 3.

## 7.2 La génération d'un conseiller

Comme nous l'avons vu, l'architecture d'un conseiller est entièrement spécifiée par le graphe des tâches. Au démarrage de l'hôte, le conseiller est "instancié" à partir du graphe des tâches et sa structure est isomorphe à celle du graphe des tâches. Cependant à cause des tâches étoiles, un conseiller n'est pas toujours entièrement généré (Fig. 6). En effet, pour ces tâches, l'instanciation "paresseuse" d'un niveau supplémentaire (le fameux "0 à n") dans le conseiller est déclenchée par la création dans un contexte donné de certains agents hôtes (Fig. 7), e.g. la création d'un *hierarchy browser* (Fig. 3). Le contexte de création est en effet tout aussi important que le nouvel agent hôte. Connaître les seules classes et méthodes entraînant la création de nouveaux agents hôtes ne suffit guère, puisque le contexte de création peut s'avérer primordial pour l'analyse. Par exemple, le contexte de la création des *browsers* en Smalltalk-80 peut donner de sérieuses indications sur les intentions d'un usager. Les tâches étoiles couplés à la détection en contexte des nouveaux agents hôtes permettent d'exprimer de telles subtilités.

La mécanique d'instanciation du graphe des agents épiphytes est basée sur un principe unique: chaque création d'un nouvel agent hôte provoque le parcours du graphe des agents épiphytes existants. Ce parcours a pour effet de connecter les agents épiphytes existant au nouvel agent hôte si nécessaire, et, le cas échéant, d'instancier les agents épiphytes étoile. Plus précisément, à chaque instanciation d'un nouvel acteur, EpiTalk effectue les deux tâches :

- il connecte les agents épiphytes existants intéressés par l'acteur;
- il instancie les tâches étoiles qui doivent l'être.

Ces deux opérations sont effectuées en même temps de la manière suivante :

Le graphe des agents épiphytes est parcouru (en profondeur d'abord) à partir du père de l'agent épiphyte qui a détecté le nouvel agent hôte. Si l'agent hôte a été créé à partir d'un bouton dans l'éditeur général, l'agent épiphyte correspondant est tout simplement l'agent épiphyte racine. Le graphe des agents épiphytes est donc entièrement parcouru. Au contraire, si la création de l'agent hôte est repérée par un agent épiphyte (terminal) déjà existant, alors uniquement le sous-graphe des tâches de cet agent épiphyte est parcouru et mis-à-jour.

Pour chacun des agents épiphytes rencontrés lors de ce parcours les opérations suivantes sont effectuées :

- (i) Si la tâche de l'agent épiphyte est une tâche terminale, et que l'agent épiphyte est intéressé par ce nouvel agent hôte, alors l'espion dédié à cet agent hôte est connecté à l'agent épiphyte. Cette connexion aura pour effet d'informer l'agent épiphyte des envois des messages identifiés dans la tâche terminale comme pertinents par le concepteur.
- (ii) Si la tâche de l'agent épiphyte est une tâche non terminale simple, alors le parcours est simplement poursuivi dans les sous-agents épiphytes.
- (iii) Enfin, si la tâche de l'agent épiphyte est une tâche étoile et que cette tâche est concernée par le nouvel agent hôte, alors la tâche est ré-instanciée totalement. et seul le nouveau sous-graphe des agents épiphytes est ensuite parcouru de manière récursive. Une tâche étoile est "concernée" par un nouvel agent hôte si au moins une de ses sous-tâches l'est (c'est donc un calcul récursif qui s'arrête aux tâches terminales).

La détection dynamique des agents hôtes posent certains problèmes d'implémentation. En effet, la représentation réflexive d'un groupe doit correspondre exactement à la structure du groupe décrit. Aussi ces créations doivent-elles être repérées afin de pouvoir greffer les espions et lier les acteurs aux agents épiphytes. Ces questions sont discutées dans [Pachet, et al., 1995].

### **7.3 L'espionnage des interactions: description et mise en oeuvre**

Dans les systèmes à objets, les interactions correspondent aux transmissions de message. De par l'encapsulation des agents hôtes, les messages sont la seule manière dont les agents peuvent interagir entre eux. Par conséquent, l'espionnage des interactions se traduit au niveau de l'implémentation par l'espionnage des transmissions de message.

Trois questions se posent alors:

- 1- Quelles sont les limites d'un système d'agents ? Quels acteurs en font partie ?  
Faut-il inclure les agents humains qui interagissent à l'aide d'une interface ?

En pratique, seules les interactions entre un usager et un système d'information qui se traduisent par un envoi de message à un agent hôte peuvent être recueillies par un système d'information. En effet, du point de vue du système d'information, seuls

les *messages transmis aux agents* après une action de l'utilisateur (et non pas directement les actions de l'utilisateur) sont accessibles. Ainsi ne pourront être observés les clics ou les actions de l'utilisateur qui ne sont pas interprétés par le système d'information hôte en termes de transmissions de message. Par conséquent dans EpiTalk, la spécification d'un espion comporte, d'une part, les sélecteurs des messages dignes d'intérêt (pour entre autres minimiser le flot de communication entre un hôte et un conseiller) et, d'autre part, les agents épiphytes à qui seront acheminés les messages observés.

2- Quels sont les éléments pertinents d'une interaction ?

Pour l'instant dans EpiTalk, la réification d'une interaction comprend l'émetteur, le destinataire, le moment, ainsi que le message transmis et le résultat rendu.

3- A quel moment, les interactions sont-elles captées ?

Les messages peuvent être observés à leur émission, pendant leur transit ou à leur réception. Actuellement EpiTalk permet de les observer à leur réception. Peu importe le moment, leur représentation et leur analyse devront résoudre les trois problèmes que sont l'ordre d'activation, le couplage des transactions et l'ordre d'arrivée des messages [Manning, 1987].

Les messages à espionner pour le compte des agents épiphytes terminaux sont spécifiés par le concepteur à l'aide d'une interface au moment de la description des tâches terminales (Fig. 8).

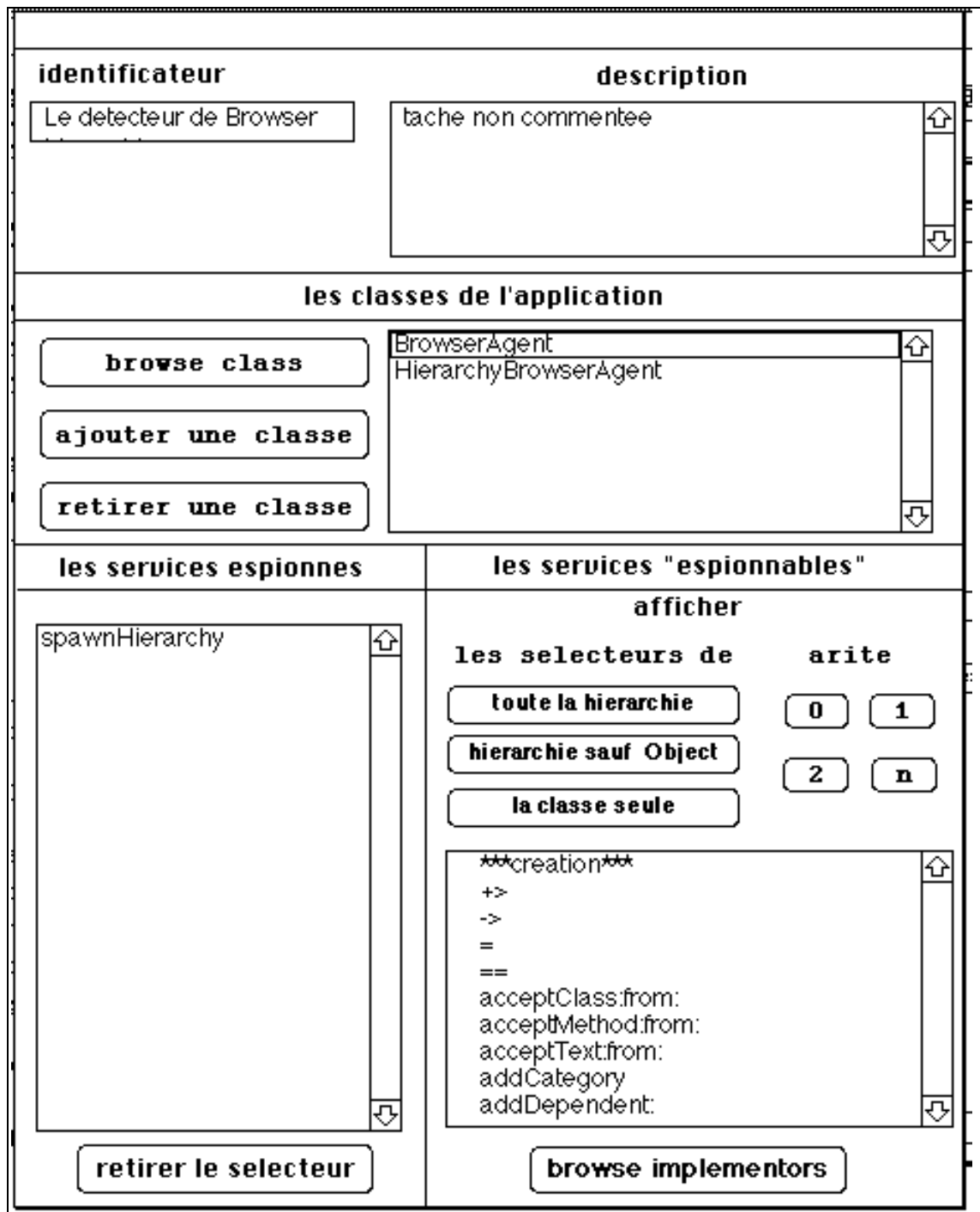


Fig. 8 L'éditeur d'espionnage.

## 8. La fonction de conseil

Dans le design de systèmes conseillers, il s'agit de trouver le bon dosage d'intervention du système en fonction des besoins de formation des apprenants. Plus le conseiller est actif ou carrément interventionniste, moins l'apprenant a de marges de manoeuvre pour

exercer son activité cognitive. A l'inverse, plus l'apprenant est actif, plus le conseiller est passif, moins il intervient. Trop de conseils et l'apprenant n'a plus d'espace pour apprendre; trop peu de conseil et la plupart des apprenants piétineront dans des voies peu productives.

Pour le moment, chaque système conseiller possède une stratégie de conseil fixe. La stratégie de conseil est dans la plupart des cas spécifiée à la conception conseil par conseil. Le concepteur décide alors

- du contenu du conseil: le conseil peut être mis en contexte en utilisant diverses valeurs d'attribut;
- de l'importance du conseil afin de discriminer lorsqu'il y a plusieurs conseils à donner;
- de la manière dont le conseil est donné: un conseil peut être donné à l'initiative du conseiller via un *PopUp Window*, à l'opposé, les conseils d'un ou plusieurs conseillers peuvent être accumulés et prodigués sur demande via une interface; certains conseils peuvent activer une alarme sonore.
- du nombre de fois qu'il est donné; un conseil peut n'être donné qu'un nombre limité de fois ou encore il peut être désactivé pour un certain temps.

Dans la phase actuelle de notre projet, nous définissons un cadre d'interprétation où s'exprimeront les différentes stratégies de conseil. Pour cela, nous implantons et expérimentons différentes stratégies de conseil. En bout de course, une stratégie sera exprimée par une base de règles NéOpus [Pachet, 92]. Diverses stratégies de conseil et cadre d'interprétation seront alors disponibles au concepteur. La stratégie de conseil pourra même évoluer dynamiquement et s'adapter à l'apprenant.

## **9. Travaux futurs : les conseillers répartis**

Les environnements informatisés de formation [Wenger 87], qui demeurent un lieu d'application privilégié des systèmes conseillers, eux aussi, n'échappent guère aux défis posés par la technologie, comme en témoignent les recherches sur les campus virtuels. Certes la formation à distance est depuis longtemps pratiquée, notamment à la Télé-Université. Cependant grâce aux nouvelles technologies, apprenants et formateurs disposeront dorénavant d'outils adéquats pour collaborer à distance. Les systèmes conseillers font partie de cet arsenal. Dès lors, les systèmes conseillers, outre la



modélisation d'un individu, la représentation de la tâche et les stratégies d'interventions, doivent investir la dimension collaborative de l'apprentissage, incontournable au plan pédagogique. Malheureusement cette dimension a été occultée jusqu'à aujourd'hui dans la majorité des conseillers développés, e.g. [Brown, et al., 82] [Shute et Glaser, 90] [Braham, et al., 92]. Cependant certains travaux commencent à considérer des axes autres que le seul couple apprenant/système, e.g. [Vivet, 91]. Un campus virtuel constitue l'exemple par excellence d'un environnement de formation réparti. Jusqu'à présent, EpiTalk a été utilisé pour le développement de systèmes conseillers dédiés à l'activité d'un seul individu [Paquette, et al., 94] [Marcenac, et al. 94] [Leman, et al., 94].

Les moyens de communication actuels ouvrent de nouvelles possibilités au niveau du travail collaboratif à distance. La formation à distance tout en tirant parti de cette technologie, ne peut plus occulter la dimension collaborative de l'apprentissage. Un des défis qui surgit alors est la construction de systèmes conseillers répartis dédiés au travail en groupe. La réalisation de tels systèmes conseillers est certes ambitieuse. Celle-ci soulève de nombreuses questions quant à la modélisation répartie des intervenants et de leur groupe de travail, la représentation de l'expertise sur la tâche et la fonction de conseil dans des groupes. Finalement, le développement de systèmes conseillers répartis demandent aussi un environnement de programmation adapté. EpiTalk contient en lui-même des solutions à ce défi. Le graphe des tâches aidera à déterminer la répartition physique des agents. Les problèmes de contrôle liés à la répartition du système conseiller seront traités à travers l'autonomie intrinsèque aux agents. Les dévermineurs répartis seront perçus comme une forme singulière de conseillers.

Pour conseiller, il faut disposer d'une part de modèles de l'objet *idéal* sur lequel portent les conseils et d'autre part de modèles de la situation passée, immédiate et future. Pour complexifier le tableau, l'objet du conseil et le but de la collaboration à l'intérieur d'un groupe peuvent adopter des visages variés. Ils peuvent n'exister qu'à travers l'ensemble du groupe seulement, e.g. le rapport entre une fourmi et sa fourmilière [Drogoul et Ferber, 91]. Les intervenants peuvent partager un but commun, e.g. produire un rapport d'équipe, ou encore chaque intervenant peut collaborer, mais conserver des buts qui lui sont propres. Ces points soulèvent comme questions:

- quel est l'objet du conseil? quel est le but de la collaboration?

- comment modéliser les transactions et l'évolution des transactions entre intervenants selon les activités d'apprentissage ou selon une tâche générique [Chandrasekaran 87]?
- comment gérer des points de vue multiples dans les conseillers ?
- comment caractériser les groupes afin de déterminer les interventions de conseils les plus pertinentes ?

Encore une fois dans EpiTalk, le graphe des tâches va servir de pivot pour articuler la modélisation de l'objet du conseil et le système conseiller comme tel. Chaque graphe représente alors un point de vue particulier sur la tâche ou sur la démarche du groupe. La décomposition hiérarchique s'est avérée adéquate dans le cadre de conseiller individuel autant au niveau de la description de l'objet à produire [Paquette, et al., 94] que de la démarche d'un individu [Paquette, et al., 93b]. En plus des liens de décomposition, le graphes des tâches peut contenir des relations de précédence (Fig. 5). A travers celles-ci, le passage d'un noeud à un autre situe les interactions dans le temps et sert à anticiper les interactions à venir. Expert et apprenant sont alors comparés à partir de modèles par recouvrement. Chaque point de vue via le graphe des tâches associé donne naissance à un système conseiller réparti autonome.

## 10. Conclusion

Dans le cadre de la formation à distance, l'absence de tutorat direct exige la mise en place de ressources pour relancer un processus d'apprentissage en difficulté ainsi que des outils pour surveiller la progression des étudiants. Les systèmes conseillers constituent une composante privilégiée de l'arsenal de support pour la formation à distance. EpiTalk est une plate-forme qui permet de décrire, de générer et de gérer des systèmes conseillers multi-agents. EpiTalk propose un certain nombre de solutions aux problèmes inhérents à la construction de systèmes conseillers. Ces solutions reposent sur l'autonomie des agents conseillers, sur le regroupement des agents en organisations et sur la structuration des ces organisations par l'intermédiaire d'un graphe des tâches. Chaque graphe des tâches correspond à un point de vue particulier. Ces points de vue cohabitent grâce à l'autonomie des agents. Le graphe des tâches agit en intermédiaire entre le concepteur et le système conseiller qui sera généré. Il articule les différentes connaissances sur le conseil. Ses informations permettront au conseiller d'anticiper les actions d'un intervenant ou du groupe; de même, ses informations permettront à un conseiller de discriminer entre plusieurs conseils et de choisir le plus pertinent. Par

analogie avec la botanique, les conseillers sont dits "épiphytes" car ils sont greffés et croissent sur des applications existantes tout en préservant leur intégrité. Actuellement différentes stratégies de conseil sont explorées et des outils conséquents sont développés et intégrés à EpiTalk.

## 11. Bibliographie

- [Abelson et Sussman, 1985] Abelson, Harold; and Gerald Jay Sussman, *Structure and Interpretation of Computer Programs*, Cambridge, Mass., The MIT Press, 1985, 542 p.
- [Brahan, et al., 92] Brahan, J. W., B. Farley, R. A. Orchard, A. Parent, C.S. Phan, *A Designer's Consultant*, Technical Paper #NRCC33234, Institute for Information Technology, NRC, Ottawa, 1992.
- [Briot, 1989] Briot, Jean-Pierre, Actalk: a Testbed for Classifying and Designing Actor Languages in the Smalltalk-80 Environment, ECOOP '89 Proceedings, Nottingham, Angleterre, 10-14 juillet 1989, pp. 109-129.
- [Brown, et al., 82] Brown, J.S., R.R. Burton et J. de Kleer, *Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II, and III*, dans Sleeman and Brown éd. Intelligent Tutoring Systems, Academic Press, London 1982.
- [Chandrasekaran 87] Chandrasekaran B. *Towards a functional architecture for intelligence based on generic information processing tasks*. Proc. of the Tenth IJCAI, Milan (Italy), Vol. 2, 1987, pp. 1183-1192.
- [Drogoul et Ferber, 91] Drogoul, Alexis et Jacques Ferber, *A Behavioral Simulation Model for the Study of Emergent Social Structures*, European Conference on Artificial Life, Toward a practice of Autonomous Systems (ECAL '91) Proceedings, 11-13 décembre 1991, Paris, France, The MIT Press/Bradford Books.
- [Foote et Johnson, 1989] Foote, Brian et Ralph E. Johnson, *Reflective Facilities in Smalltalk-80*, OOPSLA Proceedings, 1-6 octobre 1989, New Orleans, Louisiana, Special Issue of SIGPLAN Notices, vol. 24, no 10, octobre 1989, pp. 327-335.
- [Giroux et al. 94] Giroux, Sylvain, François Pachet et Gilbert Paquette, Des systèmes d'information multi-agents épiphytes, Deuxièmes Journées Francophones Intelligence Artificielle Distribuée et Systèmes Multi-Agents, Grenoble, France, 9-11 mai 1994, pp. 211-222.
- [Giroux, 1993] Giroux, Sylvain, *Agents et systèmes, une nécessaire unité*, thèse de Ph. D. en informatique, Département de mathématiques et de recherche opérationnelle, Université de Montréal, août 1993, 246 p.
- [Giroux, et al. 94b] Giroux, S., F. Pachet et J. Desbiens, *Debugging Multi-Agent Systems: a Distributed Approach to Events Collection and Analysis*, CWDAI'94, Canadian Workshop on Distributed Artificial Intelligence, Banff, Canada, May 16, 1994.
- [Hiltz, 86] Hiltz, S. R., *The Virtual Classroom: Using Computer-Mediated Communication Systems for University Teaching*. Journal of Communication, 1985, pp. 95-104.
- [Hiltz, 90] Hiltz, S. R., *Evaluating the Virtual Classroom* in [Harasim, 90].
- [Leman, et al., 94] Leman, Stéphane, Sylvain Giroux et Pierre Marcenac, *A Generic Distributed Method for Cognitive Modelling*, AI '94, 7th Australian Joint Conference on Artificial Intelligence, November 1994.
- [Manning, 1987] Manning, Carl R., *Traveler: The Apiary Observatory*, ECOOP Proceedings, 1987, Paris, 15-17 juin, pp. 97-105.
- [Marcenac, et al. 94] Marcenac, Pierre, Sylvain Giroux et Stéphane Leman, *A Multi-Agent Approach to Student Modelling*, EW-ED'94: Third East-West Conference on Computer Technologies in Education, 19th-23rd Sept. 1994, Yalta, Crimea, Ukraine.
- [Pachet, 1992] Pachet, François, *Représentation de connaissances par objets et règles: le système NéOpus*, thèse de doctorat, LAFORIA, Université Pierre et Marie Curie, France, septembre 1992.

- [Pachet, et al. 95] F. Pachet, F. Wolinski et S. Giroux, *Spying as an object-oriented programming paradigm*, TOOLS 95, 6-9 mars 1995, Versailles, France, à paraître.
- [Paquette 92]. Paquette, G. *Metaknowledge in the LOUTI development system*. Proc. of CSCSI-92, Vancouver, Canada, May 1992.
- [Paquette, et al., 93b] Paquette, Gilbert, Françoise Crevier, Sylvain Giroux et François Pachet, *Méthodes et outils de développement de systèmes conseillers dans les environnements de formation*, Rapport de recherche, LICEF, Télé-Université, Montréal, déc. 93.
- [Paquette, et al., 93] G. Paquette, G. Bergeron et J. Bourdeau, *The Virtual Classroom revisited*, Conference TeleTeaching'93, Trondheim, Norvège, août 1993.
- [Paquette, et al., 94] Paquette, G., et al., *Design of a Knowledge-based Didactic and Generic Workbench*, CALISCE '94, August 31, September 1-2 1994, Paris, France, pp. 303-311.
- [Shute et Glaser, 90] Shute, V. J. et R. Glaser, *A Large-Scale Evaluation of an Intelligent Discovery World: Smithtown*, Interactive Learning Environments, vol. 1, no 1, Ablex, mars 1990.
- [Smith, 1983] Smith, Brian Cantwell, *Reflection and Semantics in Lisp*, Proceedings of the 1984 ACM Principles of Programming Languages Conference Proceedings, 1983, pp. 23-35.
- [Sowa, 84] Sowa, John F., *Conceptual Structures*, Addison-Wesley, 1984, 481 p.
- [Vivet 91] Vivet, M. *Knowledge Based Systems for Educations: Taking into account the Learner's Context*, PEG-91, Genève, Suisse, mai 91.
- [Wenger 87] Wenger, E., *Artificial intelligence and Tutoring Systems, Computational and Cognitive Approaches to the Communication of Knowledge*, Morgan-Kaufman, Los Altos, CA, 1987.
- [Wilson & Jonassen 90] Wilson B.G. Jonassen, D.H. *Automated instructional systems design: A review of prototype systems*. Journal of Artificial Intelligence in Education, vol. 2, no 2, pp. 17-30, 1990.