

ACADÉMIE DE MONTPELLIER
UNIVERSITÉ MONTPELLIER II
- SCIENCES ET TECHNIQUES DU LANGUEDOC -
DIPLOME D'ÉTUDES APPROFONDIES
- INFORMATIQUE -

*Modélisation de l'acquisition de
connaissances par le dialogue entre
agents cognitifs*

MEHDI YOUSFI MONOD

Date de soutenance : 26 JUIIN 2003

Directeur de stage : VIOLAINE PRINCE

REMERCIEMENTS :

Je remercie particulièrement madame Violaine Prince pour son aide et sa patience face à mes nombreuses questions tout au long de ce stage. Je souhaite également remercier Didier Schwab pour ses conseils éclairés qui se sont révélés très utiles. J'exprime enfin ma reconnaissance à Jacques Chauché et Mathieu Lafourcade pour leur sympathie et leur écoute.

Table des matières

1	Introduction	2
2	L'acquisition de connaissances par le dialogue entre agents cognitifs	3
2.1	Les agents cognitifs	3
2.2	L'apprentissage par le dialogue	4
2.2.1	L'apprentissage hors dialogue	4
2.2.2	Les modèles de dialogue	4
2.3	Le raisonnement	5
3	La théorie	6
3.1	Les agents	6
3.2	La base de connaissances	6
3.3	Le dialogue	8
3.4	L'apprentissage	11
3.4.1	Apprendre les données fournies par le maître	11
3.4.2	Les élèves curieux	14
4	Architecture et conception	16
4.1	Architecture	16
4.2	Conception	18
4.3	Les classes java	19
4.4	Les algorithmes de gestion du conflit	19
5	Mise en oeuvre et évaluation	22
5.1	Premier exemple : un conflit d'implications	22
5.2	Deuxième exemple : un conflit de faits	23
6	Conclusion et perspectives	25

Chapitre 1

Introduction

Ce travail se situe dans le cadre de la communication entre agents. Les agents que nous considérons peuvent être naturels ou artificiels.

Un agent naturel est un être vivant. Un agent artificiel est un artefact, une création artificielle créée dans le but d'aider l'homme.

Les agents se distinguent aussi sous deux autres catégories :

- les agents cognitifs : pourvus de connaissances, de modes de dérivation des connaissances et de buts
- les agents réactifs : dotés principalement de mécanismes déterministes d'action — réaction

Nous nous intéressons ici aux agents artificiels cognitifs.

Nous utilisons la communication dans le but de transmettre des connaissances entre agents. Nous souhaitons nous rapprocher du mieux que possible d'une situation naturelle de communication entre êtres humains afin de bénéficier de l'expérience de l'Homme dans le domaine de l'apprentissage. C'est donc par un dialogue que nos agents communiqueront.

D'après Kerbrat-Orecchioni [7], le dialogue est : "Parler, c'est sans doute échanger des informations ; mais c'est aussi effectuer un acte régi par des règles précises qui prétend transformer la situation du récepteur et modifier son système de croyances et/ou son attitude comportementale ; corrélativement, comprendre un énoncé c'est comprendre son contenu informationnel, sa visée pragmatique ?". Le dialogue permet en cas d'incompréhension totale ou partielle de demander des précisions, des exemples ou des explications sur le contenu d'un message reçu.

Nous nous focalisons sur le dialogue mettant en jeu deux agents, l'un désireux d'enseigner des connaissances, l'autre désireux d'en apprendre. Le choix du dialogue comme moyen d'apprentissage a été fait car c'est une méthode efficace, rapide, bien connue des hommes et peu traitée en recherche informatique.

Dans le chapitre 2 nous définirons précisément la nature de nos agents, les caractéristiques nécessaires qu'ils doivent posséder. Nous spécifierons quel langage ils utilisent et quels types de connaissances ils possèdent. Nous verrons ensuite l'état actuel des recherches sur l'apprentissage avec ou sans dialogue. Puis nous présenterons quelques modèles de dialogues existants. Nous préciserons celui que nous avons choisi et pour quelles raisons. Nous terminerons cette partie en discutant des capacités de raisonnements de nos agents. Le chapitre 3 décrit en détails notre théorie sur l'apprentissage par le dialogue entre agents cognitifs. L'architecture de notre système et l'implémentation d'une partie de notre théorie dans un programme informatique seront décrites dans le chapitre 4. Dans le chapitre 5 nous discuterons des résultats de la mise en oeuvre. Enfin nous terminerons avec le chapitre 6 qui résumera les apports de notre contribution et proposera quelques perspectives.

Chapitre 2

L'acquisition de connaissances par le dialogue entre agents cognitifs

Notre but est de définir un ensemble d'algorithmes d'acquisition de connaissances par le dialogue entre agents. Un agent est [5] " une entité physique ou virtuelle :

- qui est capable d'agir dans un environnement,
- qui peut communiquer directement avec d'autres agents,
- qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
- qui possède des ressources propres,
- qui est capable de percevoir (mais de manière limitée) son environnement,
- qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
- qui possède des compétences et offre des services
- qui peut éventuellement se reproduire,
- dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit."

Étant dans une situation d'apprentissage nous avons choisi un échange d'information sous forme de dialogue socratique (maître — élève).

L'agent " élève " est muni de mécanismes de révision de croyances et de tous les axiomes permettant le raisonnement formel. Afin d'éviter l'ambiguïté intrinsèque des langues naturelles, les données des messages seront échangées dans le langage de la logique des prédicats du premier ordre incluant les fonctions. Ce choix a été pris car le traitement de la langue naturelle ne fait pas partie du sujet et il n'existe encore aucun analyseur sémantique qui puisse réaliser une analyse satisfaisante pour nos objectifs. Cependant la situation de dialogue restera naturelle dans le sens où elle sera calquée du mieux que possible sur celle d'un dialogue homme-homme, entre un maître et un élève.

Nous considérons que les agents usent d'un formalisme commun. En revanche, l'agent " élève " peut ne pas connaître des prédicats ou fonctions proposées par l'agent " maître ", et peut l'interroger à ce sujet avant de réviser sa base.

2.1 Les agents cognitifs

Nos agents possèdent une base de connaissances et tentent de la faire évoluer par le dialogue ou par eux-mêmes. Ils doivent donc être des agents cognitifs, c'est à dire des agents bénéficiant :

- de connaissances : sur leur état, sur celui des autres agents et sur l'environnement. Nos

- agents peuvent ainsi connaître des faits et des règles qu'ils pensent être vraies dans cet environnement,
- de modes d'acquisition de connaissances : leur permettant d'incorporer à leur base des éléments de connaissances provenant de l'extérieur. Notre cas traitera uniquement de l'acquisition par le dialogue, donc ce seront des interactions entre agents et pas entre agent et environnement,
 - de modes de dérivation des connaissances : afin qu'ils puissent faire évoluer leurs connaissances en utilisant uniquement leurs propres données. Ils peuvent aussi réviser leur base afin d'éliminer d'éventuelles incohérences. Ce mode d'acquisition est le raisonnement,
 - d'un ou plusieurs buts : indispensables pour guider leurs actions jusqu'à satisfaction.

2.2 L'apprentissage par le dialogue

Durant notre recherche sur les travaux portant sur l'apprentissage et le dialogue, aucun article traitant précisément de l'apprentissage d'agents cognitifs par le dialogue n'a été trouvé. Ce peut donc être un sujet encore non exploré par la recherche. Cependant nous allons voir (section 2.2.1) qu'il existe de nombreux articles utilisant des méthodes d'apprentissage sans dialogue. Nous verrons ensuite (section 2.2.2) quels types de modélisation de dialogue existent et quel a été notre choix.

2.2.1 L'apprentissage hors dialogue

L'apprentissage est appelé, en informatique, le "Machine Learning". Il existe une grande variété de méthodes d'apprentissage dont :

- l'apprentissage par renforcement [10]
Il utilise un système à base de récompenses, en fonction de la réussite ou de l'échec d'actions effectuées par l'agent. Une base de décision en fonction d'un état donné est alors construite. Dans notre cas il serait difficile d'estimer un échec ou une réussite suite à l'apprentissage d'une règle ou d'un fait. De plus cette méthode n'est utilisable que dans des cas où le nombre de possibilités d'actions de l'agent est suffisamment restreint pour pouvoir être circonscrit par un nombre fini et raisonnable de réussites ou d'erreurs. Or, dans un dialogue le nombre de messages possibles tend vers l'infini.
- l'apprentissage supervisé [16]
Dans ce cas l'agent est au préalable instruit grâce à un ensemble de situations d'entraînement. Ces dernières sont connues et analysées par un superviseur qui sait comment réagir face à ces situations et il peut donc en informer l'agent au préalable. Ainsi l'agent aura appris comment réagir face à certaines situations typiques. Puis il est censé, avec uniquement ces acquis, faire face à toutes les éventualités futures. Évidemment il n'existe aucune situation typique de dialogue pouvant préparer un agent pour un dialogue avec un autre car chaque agent dialoguant possède un savoir quelconque et donc imprévisible.
- les réseaux de neurones [13] [2] [15]
Le principe étant basé sur une technique d'essais / erreurs est incompatible avec notre travail pour les mêmes raisons que l'apprentissage par renforcement : les essais / erreurs ne suffisent pas à recouvrir l'ensemble des possibilités de dialogues envisageables.

2.2.2 Les modèles de dialogue

La plupart des modèles de dialogue en informatique sont fondés sur la notion de *plan* [1] [4] et s'appuient sur la théorie des actes de langage d'Austin [3], reprise par Searle [14], pour représenter le dialogue comme une suite d'actions communicatives dont le but est de modifier

l'état mental des agents participants. Nos agents possèdent des buts (enseigner ou apprendre une leçon) mais n'ont pas de plans prédéterminés : il réagissent au *coup par coup*, en fonction de la réponse de l'interlocuteur. C'est pourquoi nous avons choisi un modèle *opportuniste* des actions langagières plutôt qu'un modèle de planification des actions langagières [8]. Nous sommes dans une situation de dialogue finalisé (car régit par une tâche à accomplir) et selon Luzzati ce type de dialogue se structure selon deux axes [9] :

“Le dialogue peut s'orienter dans deux directions : soit demande d'information et délivrance des renseignements s'enchaînent sans difficulté, et il s'agit d'un dialogue *régissant* ; soit des demandes de précision, d'explication, de confirmation ou de reformulation doivent intervenir pour qu'une question ou une réponse soit acceptée, et il s'agit d'un dialogue *incident*.”

La nature d'un cours nous place naturellement dans le cas d'un dialogue incident. Nous avons opté pour la modélisation des rôles fonctionnels [11] de Violaine Prince car elle nous permet de gérer un tel dialogue.

Nos échanges s'effectuant entre agents artificiels, nous avons dû adapter cette modélisation qui est plutôt orientée dialogue homme-homme ou homme-machine.

2.3 Le raisonnement

Nos agents, plus particulièrement l'agent élève, vont devoir apprendre des connaissances. Le raisonnement leur permettra de comprendre des connaissances difficiles ou de créer de nouvelles connaissances à partir d'autres déjà connues.

Le raisonnement est un mode de dérivation des connaissances. Raisonner modifie l'état de l'agent et ce sont les étapes et les résultats du raisonnement qui permettent à l'agent d'apprendre.

Notre méthode s'inspire des comportements humains qui utilisent trois modes de dérivation des connaissances [12] :

- la déduction consiste à créer de nouvelles connaissances à partir d'autres déjà connues, son mécanisme repose sur la propriété de transitivité de l'opérateur “implication”
- l'induction, par laquelle on crée des lois à partir de similitudes remarquées entre faits observés,
- l'abduction qui est un procédé naturel d'apprentissage constamment utilisé par les hommes qui est souvent vu comme le procédé qui consiste à trouver la meilleure cause possible à un fait, une observation, un événement. De manière plus générale nous considérerons que c'est associer un fait observé à une règle connue.

Chapitre 3

La théorie

3.1 Les agents

Nous utilisons un environnement multi-agents, composé uniquement d'agents artificiels cognitifs. Tout agent dans cet environnement est potentiellement maître ou élève. Le seul moyen d'interaction entre les agents est le dialogue. Nous tentons de nous rapprocher du mieux que possible d'une situation naturelle de dialogue entre des êtres humains.

Nous nous focalisons sur le dialogue mettant en relation deux agents. Des échanges de ce type peuvent se dérouler entre n'importe quel agent de l'environnement. Toutefois nous ne permettons pas à un agent élève de participer à deux dialogues simultanés avec deux maîtres car l'apprentissage serait alors malaisé.

Durant cette relation, un agent jouera le rôle du maître et l'autre de l'élève et ils garderont ce statut tout le long du dialogue. L'objectif est de faire apprendre à l'agent élève des connaissances et des informations par l'agent maître, c'est donc une situation de cours. L'attribution des rôles est cependant momentanée car elle dépend de la tâche à réaliser et des compétences de chaque agent. Dans notre cas la tâche est le cours. L'agent maître doit alors posséder les compétences nécessaires pour pouvoir enseigner à l'agent élève.

Par la suite nous utiliserons les termes d'élève et de maître pour désigner respectivement l'agent élève et l'agent maître. Le maître a pour but d'enseigner un ensemble prédéterminé de données à l'élève.

Nous faisons l'hypothèse que nos agents sont coopératifs. Ainsi aucune information erronée ne sera échangée et les agents tenteront, par tous les moyens dont ils disposent, de satisfaire les attentes de l'autre. Toutefois l'élève peut, comme dans une situation naturelle, être peu motivé et de ce fait rendre plus difficile la tâche du maître. L'élève pourra par exemple rester peu précis sur ses problèmes d'incompréhension.

3.2 La base de connaissances

Chaque agent dispose d'une base de connaissances (BC). Son contenu est dans le langage de la logique des prédicats du premier ordre, chaque connaissance d'un agent est donc une formule. Toute formule correspond à une connaissance humainement compréhensible. Nos agents élèves, comme les êtres humains, peuvent se tromper et le maître est là pour leur apprendre "ce qui est juste"¹.

Chaque base est construite manuellement. C'est son évolution qui sera automatique.

Afin de simplifier la modélisation nous utiliserons uniquement des formules de type :

– (P) ,

¹Les maîtres, naturels ou artificiels, peuvent certes se tromper sur la véracité de certaines connaissances cependant notre élève prendra pour vrai tout ce que le maître lui enseignera.

- $(P \rightarrow Q)$,
- $(P \leftrightarrow Q)$,

avec (P) et (Q) des conjonctions de prédicats (ou leur négation) d'arités quelconques et de la forme $(p(A))$ ou $(p(X))$ (ou $(\text{non}(p(A)))$ ou $(\text{non}(p(X)))$), avec $A = \{A1, A2, \dots, AN\}$ un ensemble de termes et $X = \{X1, X2, \dots, XN\}$ un ensemble de variables. Nous noterons par la suite P et Q de telles conjonctions de prédicats. Une disjonction de prédicats sera séparée en plusieurs formules :

$(p(x) \cup q(x) \rightarrow r(x))$ sera noté $(p(x) \rightarrow r(x))$ puis $(q(x) \rightarrow r(x))$

Pour qu'un apprentissage puisse avoir lieu, un pré-requis nous paraît nécessaire : avant le dialogue, nos deux agents possèdent déjà des connaissances — nous les appellerons connaissances de base — et afin de se comprendre sur un minimum de points, les deux agents doivent avoir un certain nombre de connaissances identiques, c'est-à-dire qu'une partie de leur BC possèdera les mêmes formules. Ces connaissances servent de base pour l'apprentissage : le maître doit s'appuyer sur des connaissances connues des deux agents afin d'en expliquer de nouvelles à l'élève. Cependant il n'y a pas de BC partagée, celles des agents sont bien distinctes.

Lors de l'apprentissage, chaque agent va tenter de rendre sa BC la plus "connexe" possible.

Est appelée *connexe* une base qui vérifie ces deux conditions :

- $(\forall p(A) \text{ ou } \forall p(X), \exists P \rightarrow Q / p \in P \cup Q)$.
- $(\forall F = (P \rightarrow Q \text{ ou } P \leftrightarrow Q))$, soit tous les prédicats de (P) , soit tous les prédicats de (Q) vérifient la première condition dans l'ensemble des formules privé de (F) .

Ainsi dans une BC connexe chaque élément de connaissance a un lien plus ou moins direct avec chaque autre élément. Nous cherchons à calquer la situation de dialogue entre ces agents artificiels avec une situation naturelle entre êtres humains, or un être humain peut souvent — mais pas toujours — faire le lien entre deux de ses connaissances prises au hasard. Le chemin entre ces deux dernières peut être long mais il peut exister.

L'être humain sait par exemple classer ses connaissances par la relation d'hyponymie — hyponymie ("est un") ce qui lui permet de faire le rapprochement entre deux objets très différents : ce sont tout de même des objets, ils vérifieront donc les propriétés associées aux objets. Cependant s'il essaie de faire le lien entre *un parapluie* et *la bonté*, il se peut qu'il n'y parvienne pas ou que le lien qu'il trouve soit très subjectif.

C'est pourquoi, dans un souci de ressemblance avec une situation naturelle, nos agents ne possèderont forcément pas des connaissances de base connexes. Ceci nous permettra de leur donner des connaissances de base assez variées sans être obligé d'établir un lien entre chacune.

Nos agents se restreindront alors à garder seulement une connexité entre l'ensemble des connaissances de base, noté $\Omega = \{\omega_1, \omega_2, \dots, \omega_n\}$ avec chaque ω_i étant une composante connexe, et toute nouvelle connaissance apprise ψ . Ce qui signifie que ψ va "s'attacher" à un ω_i de telle sorte que l'ensemble de connaissances $\{\psi\} \cup \omega_i$ soit connexe. Il se peut que ψ s'attache simultanément à deux composantes ω_i et ω_j créant ainsi une plus grande composante connexe composée de ψ , ω_i et ω_j .

Par exemple si le maître veut enseigner à l'élève que "les êtres humains sont mortels" (1), il est préférable que l'élève connaisse au moins la notion d'"être humain" ou celle de "mortel"². Dans le cas contraire il aurait pu acquérir de manière brute cette connaissance mais n'aurait pu l'utiliser en relation avec les autres connaissances de sa base ou du moins pas tant que de nouvelles formules implicatives n'auraient relié cette nouvelle connaissance aux autres.

En revanche s'il possédait une connaissance comme "les mortels sont vivants" alors l'ajout de la formule (1) conserverait la connexité des composantes de la base et l'élève pourrait, de sa

²Cependant l'élève n'est pas obligé de connaître explicitement l'une de ces deux notions : il lui suffit de posséder toutes les connaissances nécessaires à la compréhension d'une. Nous étudierons ce cas dans la section 3.4.

base et de la nouvelle règle, déduire que “les êtres humains sont vivants”. Toutefois la connexité peut être conservée sans pouvoir tirer des déductions des nouvelles connaissances : si l’agent sait uniquement que “les êtres humains sont mortels” alors en apprenant que “les êtres humains sont intelligents” il ne pourra rien déduire de plus que ces deux implications. Mais il pourrait vouloir hypothétiser (section 3.4) : est-ce qu’être intelligent entraîne le fait d’être mortel (ou réciproquement) ?

Nous comprenons alors que si les deux agents ne possèdent pas des connaissances communes alors il sera impossible de conserver la connexité des composantes connexes de la BC de l’élève.

3.3 Le dialogue

Un dialogue entre nos deux agents s’engage. Le maître doit savoir ce qu’il va enseigner à l’élève. Nous le lui avons spécifié au préalable sous forme de plusieurs éléments de cours, chacun contenu dans une formule logique. Le maître va ensuite fournir chaque formule à l’élève en attendant entre chacune que l’élève lui réponde qu’il l’a correctement comprise ou non.

L’élève répondra alors par :

- une confirmation de compréhension, le maître pourra alors continuer son cours,
- une simple incompréhension,
- une demande spécifique sur un élément précis.

Un dialogue s’installe alors. Nous utilisons la théorie des rôles fonctionnels comme modélisateur des échanges de ce dialogue. Cette théorie nous permet d’attacher un rôle à chaque intervention. Ainsi les deux agents, en recevant un message, connaissent son rôle et peuvent fournir une réponse adéquate.

Dans les types de rôles fonctionnels nous utilisons les notions de connaissance, information, exemple, explication et précision — rassemblées sous le terme de donnée — que nous définissons ainsi :

- *connaissance* : c’est une formule logique générale (faisant intervenir une variable) universelle ou existentielle.

Ainsi ($humain(x) \rightarrow mortel(x)$) est une formule universelle : “les êtres humains sont mortels” et ($humain(x)$) est une formule existentielle : “la notion/le prédicat être humain existe et il y a des valeurs rendant à vrai ce prédicat, c’est-à-dire il existe des êtres humains”

- *information* : Elle sert principalement de réponse et prend donc comme valeurs possibles soit {vrai, faux}, soit un terme, soit “inconnu”.

Par exemple ($humain(Jean)$) s’évalue à vrai ou faux : “Jean est un être humain” ou “Jean n’est pas un être humain”, ici l’information est la réponse à la question “Jean est-il un être humain?”. Tout comme ($humain(x) \rightarrow mortel(x)$) : “les êtres humains sont mortels” ou “les êtres humains ne sont pas mortels”.

Dans le cas d’un terme, c’est une fonction qui sert d’évaluateur, c’est-à-dire qui fournit l’évaluation d’une caractéristique d’une ou de plusieurs constantes. Nous pouvons par exemple évaluer la valeur d’une constante avec la fonction ($val(x)$) qui représente la valeur numérique de son argument, l’information est cette valeur.

- *exemple* : c’est le résultat de l’instanciation d’une formule contenant au moins une variable.
- *explication* : c’est une donnée dépendant du type de demande.
 - si la demande porte sur un prédicat, alors la réponse sera une ou plusieurs équivalences entre une formule logique conjonctive et le prédicat,
 - si la demande porte sur une fonction f, alors la réponse sera un prédicat binaire d’égalité prenant en argument une composée de fonctions et f,

- si la demande porte sur une implication de la forme $(P \rightarrow Q)$, alors la réponse sera une série d'implications montrant le cheminement de (P) à (Q) .
- *précision* : c'est une donnée qui intervient lorsqu'une information fournie ne satisfait pas totalement l'agent et qu'il souhaite aller davantage dans les détails.

Voici les différents types de rôle fonctionnels. Ils sont illustrés d'exemples typiques contenant une question (demande) puis une réponse (donne) à cette question :

1. demande/donne-connaissance :

- demande-connaissance : En général il n'y a pas de demande explicite de connaissance : il est peu habituel, dans un dialogue "naturel", de poser une question qui aura pour réponse, par exemple, "les êtres humains sont mortels" (2). En effet si l'élève demande "Les êtres humains sont-ils mortels?" alors la réponse sera de type booléenne (oui ou non) et non la phrase affirmative (2).
- donne-connaissance : Ce type est principalement utilisé pour enseigner une connaissance, il introduit un échange avec l'élève.
Forme générale : $(P \rightarrow Q)$ ou $(P \leftrightarrow Q)$ avec au moins une variable.
Exemple : *donne – connaissance*(*home*(x) \rightarrow *mortel*(x)) : "Les êtres humains sont mortels."

2. demande/donne-information :

- évaluation booléenne :
demande-information :
Forme générale : soit (P) , soit $(P \rightarrow Q)$, soit $(P \leftrightarrow Q)$, avec ou sans variable.
Exemples :
demande – information(*humain*(*Jean*)) : "Jean est-il un être humain?"
demande – information(*humain*(x) \rightarrow *mortel*(x)) : "Les êtres humains sont-ils mortels?"
donne-information :
Forme générale : soit (*Vrai*), soit (*Faux*), soit (*Inconnu*)
Exemple : *donne – information*(*Vrai*) : "Oui."
- évaluation par fonction :
demande-information :
Forme générale : soit *egal*($f(a), x$) avec *egal* le prédicat d'égalité, f la fonction d'évaluation et a la constante à évaluer, soit (*Inconnu*)
Exemple : *demande – information*(*egal*(*val*(g), x)) : "Quelle est la valeur de g ?"
donne-information :
Forme générale : *egal*($f(a), r$), avec r la constante réponse
Exemple : *donne – information*(*egal*(*val*(g), 9.81)) : "La valeur de g est 9.81."

3. demande/donne-exemple :

- demande-exemple :
Forme générale : (P) avec au moins une variable.
Exemple : *demande – exemple*(*humain*(x)) : "Citez-moi un être humain"
- donne-exemple :
Forme générale : soit (P) avec toutes les variablesinstanciées, soit (*Inconnu*)
Exemple : *donne – exemple*(*humain*(*Jean*)) : "Jean est un être humain."

4. demande/donne-explication :

- pour un prédicat :
demande-explication :
 Forme générale : $(p(x))$, avec p un prédicat
 Exemple : *demande – explication*(*humain*(x)) : “Qu’est-ce qu’un être humain?”
donne-explication :
 Forme générale : soit $(p(x) \leftrightarrow P)$, soit (*Inconnu*)
 Exemple : *donne – explication*(*humain*(x) \leftrightarrow (*animal*(x)*etintelligent*(x)*etparle*(x))) :
 “Un être humain est un animal intelligent qui parle.”

- pour une fonction :
demande-explication :
 Forme générale : $(f(X))$, avec f la fonction à expliquer
 Exemple : *demande – explication*(*acceleration*(x)) : “Qu’est-ce que l’accélération d’un objet?”
donne-explication :
 Forme générale : soit *egal*($f(X), g(X)$), avec g une composée de fonctions équivalente à f , soit (*Inconnu*)
 Exemple : *donne – explication*(*egal*(*acceleration*(x), *augmentation*(*vitesse*(x)))) : “L’accélération est l’augmentation de la vitesse de l’objet.”

- pour une implication :
demande-explication :
 Forme générale : $(P \rightarrow Q)$
 Exemple : *demande – explication*(*humain*(x) \rightarrow *mortel*(x)) : “Pourquoi les êtres humains sont-ils mortels?”
donne-explication : Forme générale : soit $(P_1 \rightarrow P_2)$ puis $(P_2 \rightarrow P_3)$... puis $(P_{n-1} \rightarrow P_n)$, soit (*Inconnu*)
 Exemple :
donne – explication(*humain*(x) \rightarrow *vivant*(x))
donne – explication(*vivant*(x) \rightarrow *mortel*(x))
 “Les êtres humains sont vivants et les vivants sont mortels.”

- 5. demande/donne-précision : intervient systématiquement après un échange source d’une demande de précision :
 Échange source :
demande – information(*constante*(g)) : “ g est-elle une constante?”
donne – information(*Vrai*) : “Oui.”
 Forme générale : comme pour une demande d’information avec fonction d’évaluation

- 6. dit-satisfaction : indique à l’autre agent que la dernière donnée fournie a bien été comprise.

- 7. dit-insatisfaction : indique à l’autre agent que la dernière donnée fournie n’a pas été correctement comprise.

Nous n’utilisons pas le type demande/donne-reformulation car il est caractéristique des langues naturelles : dans un dialogue naturel une personne peut mal s’exprimer — par des fautes de grammaire, de vocabulaire, ... — ou utiliser un langage trop soutenu ou pas assez ce qui résulte dans tous ces cas à une incompréhension de type “formulation”. Une reformulation peut alors souvent aider à la compréhension. Cependant, avec des agents artificiels, en utilisant un

langage basé sur la logique, ces problèmes sont contournés et la seule reformulation alors possible est de trouver une équivalence de formule. Le type demande/donne-explication fournit déjà une équivalence de formule.

3.4 L'apprentissage

L'apprentissage peut avoir plusieurs buts différents :

- enrichir la BC en apprenant de nouvelles données enseignées par le maître,
- augmenter la connexité de la BC en ajoutant des liens entre les données déjà connues,
- acquérir de nouvelles constantes en élargissant la base des prédicats, demandant des d'exemples ou des valeurs de constantes,
- comprendre pourquoi certaines formules en impliquent d'autres.

Dans notre situation de cours, le premier but est indispensable pour retenir la leçon enseignée, car elle est censée apporter à l'élève des données qu'il ne possède pas. Nous traiterons d'abord de ce cas.

Les trois autres buts sont réservés aux élèves curieux car ils ne sont pas indispensable pour comprendre le cours, nous en discuterons dans la partie "Les élèves curieux".

3.4.1 Apprendre les données fournies par le maître

Pour pouvoir apprendre, l'élève doit d'abord comprendre les données reçues. Nous utilisons "comprendre" au sens de "conserver la connexité des composantes de la BC" : l'élève comprend une donnée parce qu'elle est reliée à au moins une autre de sa BC.

Pour savoir s'il a compris une donnée, l'élève doit donc vérifier :

- pour une formule (P) qu'il connaît l'ensemble des prédicats de (P),
- pour une formule ($P \rightarrow Q$) ou ($P \leftrightarrow Q$) qu'il connaît l'ensemble des prédicats de (P) ou de (Q).

Si ce n'est pas le cas alors il va devoir informer le maître de l'incompréhension.

Dans le premier cas il suffit d'expliquer à l'élève ce qu'est P . Dans le second, il faut choisir s'il faut expliquer P ou Q , sachant que l'un des deux n'est certainement pas explicable. En effet le principal but du cours est d'accroître la BC de l'élève et il faut donc respecter la condition suivante : l'une des deux formules doit être nouvelle, afin d'apporter une nouveauté à la BC, et l'autre connue, afin de conserver la connexité des composantes de la BC.

Face à l'incompréhension dans le second cas, deux possibilités s'ouvrent alors à l'élève :

- exprimer une simple incompréhension, laissant le maître libre de choisir d'expliquer P ou Q (situation (a)),
- ou préciser ce s'il souhaite connaître P ou Q .

Le maître doit maintenant expliquer à l'élève P ou Q , nous l'étudierons dans le paragraphe "La stratégie de dialogue".

Une fois les données comprises, l'élève peut se rendre compte qu'elles sont contradictoires avec sa BC. Il va devoir trouver une solution pour gérer ce conflit comme nous le montrerons dans le paragraphe "La gestion des conflits".

Après tout conflit géré, l'élève peut apprendre la donnée fournie par le maître. De plus si c'est un élève motivé, il pourra utiliser cette donnée fraîchement acquise et sa BC pour générer de nouvelles connaissances comme nous le verrons dans la partie "Les élèves curieux".

La stratégie de dialogue : Elle a pour but de résoudre le problème d'incompréhension d'une formule par un choix judicieux de questions et de réponses. Nous adoptons une technique inspirée de la méthode socratique :

Pour chaque prédicat p_i à expliquer, le maître connaît un autre prédicat p_j qui est en relation avec p_i par une implication ou équivalence F . Donc pour s'assurer que l'élève comprenne p_i grâce à p_j , le maître va interroger l'élève pour savoir s'il connaît le prédicat p_j . Si l'élève le connaît alors il suffit au maître de donner la formule F à l'élève. Sinon le maître va devoir chercher une autre formule qui explique le prédicat p_j et ainsi de suite.

Si le maître ne parvient pas à trouver un prédicat qui explique p_i alors c'est qu'il se trouvait dans la situation (a) et qu'il a choisit le mauvais ensemble de prédicats. Il ne lui reste plus qu'à choisir l'autre.

Voici un exemple d'un dialogue avec une incompréhension :

>Maître - *donne – connaissance*($humain(x) \rightarrow mortel(x)$)
; “Les êtres humains sont mortels.”
; l'élève ne connaît aucun de ces deux prédicats, il connaît cependant les prédicats ($animal(x)$), ($intelligent(x)$) et ($parle(x)$).
>Élève - *dit – insatisfaction*()
; “Je ne comprends pas.”
; le maître choisit d'expliquer à l'élève ce qu'est un être humain plutôt que ce qu'est un mortel.
>Maître - *demande – information*($animal(x)$)
; “Sais-tu ce qu'est un animal ?”
>Élève - *donne – information*(*Vrai*)
; “Oui.”
>Maître - *demande – information*($intelligent(x)$)
; “Sais-tu ce que c'est “être intelligent ?””
>Élève - *donne – information*(*Vrai*)
; “Oui.”
>Maître - *demande – information*($parle(x)$)
; “Sais-tu ce qu'est que parler ?”
>Élève - *donne – information*(*Vrai*)
; “Oui.”
>Maître - *donne – explication*(($animal(x) \text{ et } intelligent(x) \text{ et } parle(x) \leftrightarrow humain(x)$)
; “Un être humain est un animal intelligent qui parle.”
>Élève - *dit – satisfaction*
; “Je comprends.”

L'élève ayant compris ce qu'est un être humain, il peut maintenant apprendre la connaissance ($humain(x) \rightarrow mortel(x)$) en l'ajoutant simplement à sa base, la connexité est conservée.

La gestion des conflits : Nous considérons deux cas de conflits : ceux liés à une implication et ceux liés à un fait.

Le premier cas intervient typiquement lorsque l'élève possède une connaissance de type ($P \rightarrow Q$) et tente d'apprendre une autre de type ($P \rightarrow non(Q)$). L'élève doit alors tout simplement ôter la formule ($P \rightarrow Q$) de sa BC et y ajouter ($P \rightarrow non(Q)$). Il agit ainsi car nous considérons que ce sont les connaissances du maître qui sont justes et donc qui prennent le dessus sur celles de l'élève.

Cependant le conflit peut être masqué si l'élève possède les connaissances suivantes : ($P_1 \rightarrow P_2$), ($P_2 \rightarrow P_3$), ... , ($P_{n-1} \rightarrow P_n$) et tente d'apprendre ($P_1 \rightarrow non(P_n)$). Nous observons qu'il ne possède pas la connaissance ($P_1 \rightarrow P_n$) en tant que telle mais un équivalent.

Une première solution consisterait à supprimer toute la suite d'implications et de ne conserver que celle du maître. Toutefois seul un ou quelques éléments de la série peuvent être faux. Nous

avons donc choisi une méthode plus souple qui consiste à rechercher une implication fautive et à n'éliminer que celle-là. En effet il suffit d'ôter une seule implication pour supprimer le conflit.

L'élève va tenter de valider chaque implication en effectuant pour chacune une demande d'information au maître. Dès qu'une implication fautive est repérée, l'élève l'élimine et ajoute alors la nouvelle implication en toute sécurité. Toutefois si le maître ne peut valider ou invalider aucune des implications alors l'élève sera obligé d'ôter toutes les formules de la série avant d'ajouter celle du maître, ceci afin d'être sûr de ne pas conserver une contradiction.

Voici un exemple :

; l'élève possède ces connaissances :
 ; $(\text{bonmarche}(x) \rightarrow \text{rare}(x))$; "Ce qui est bon marché est rare"
 ; $(\text{rare}(x) \rightarrow \text{cher}(x))$; "Ce qui est rare est cher"
 >Maître - *donne – connaissance* $(\text{bonmarche}(x) \rightarrow \text{non}(\text{cher}(x)))$
 ; "Ce qui est bon marché n'est pas cher"
 ; cette donnée entre en contradiction avec la base de l'élève
 ; l'élève va donc tenter de localiser la ou les implications fautes
 >Élève - *demande – information* $(\text{bonmarche}(x) \rightarrow \text{rare}(x))$
 >Maître - *donne – information*(Vrai)
 >Élève - *demande – information* $(\text{rare}(x) \rightarrow \text{cher}(x))$
 >Maître - *donne – information*(Faux)
 ; l'élève comprend alors d'où vient le conflit et révisé sa base en conséquence :
 ; l'élève possède maintenant ces connaissances :
 ; $(\text{bonmarche}(x) \rightarrow \text{rare}(x))$
 ; $(\text{bonmarche}(x) \rightarrow \text{non}(\text{cher}(x)))$

Deux autres types de conflit d'implications moins facilement détectables peut aussi intervenir :

- Supposons que l'élève possède ces trois suites d'implications Δ , Θ et Λ :
 $\Delta = ((P_1 \rightarrow P_2), (P_2 \rightarrow P_3), \dots, (P_{n-1} \rightarrow P_n))$,
 $\Theta = ((P_1 \rightarrow Q_2), (Q_2 \rightarrow Q_3), \dots, (Q_{i-1} \rightarrow Q_i))$,
 $\Lambda = (Q_{i+1} \rightarrow Q_{i+2}), \dots, (Q_{n-1} \rightarrow \text{non}(P_n))$.

Si le maître donne la connaissance $(Q_i \rightarrow Q_{i+1})$ alors un conflit apparaît car d'après Δ on peut déduire P_n de P_1 et maintenant d'après Θ , $(Q_i \rightarrow Q_{i+1})$ et Λ on peut déduire $\text{non}(P_n)$ de P_1 .

- Supposons maintenant que l'élève possède ces deux suites d'implications Θ et Λ :
 $\Theta = ((P_1 \rightarrow Q_2), (Q_2 \rightarrow Q_3), \dots, (Q_{i-1} \rightarrow Q_i))$,
 $\Lambda = (Q_{i+1} \rightarrow Q_{i+2}), \dots, (Q_{n-1} \rightarrow \text{non}(P_1))$.

Si le maître donne la connaissance $(Q_i \rightarrow Q_{i+1})$ alors un conflit apparaît car maintenant d'après Θ , $(Q_i \rightarrow Q_{i+1})$ et Λ on peut déduire $\text{non}(P_1)$ de P_1 .

Ces types de conflits sont plus difficiles à détecter car il faut parvenir à localiser les deux ou trois séries d'implications en cause. L'élimination du conflit s'effectue de la même façon que dans le précédent cas : l'élève va vérifier auprès du maître la validité de chaque implication de Δ et de Θ jusqu'à en trouver une qui est fautive afin de l'éliminer.

Le second cas de conflit apparaît lorsque l'élève croit un fait vrai $(p(A))$ et que le maître lui enseigne qu'il est faux $(\text{non}(p(A)))$ ou qu'il lui donne une connaissance qui permet à l'élève de

déduire que son fait est faux. La gestion du conflit est proche de celle du conflit d'implications, plusieurs cas peuvent intervenir (nous sommes dans la situation où l'élève croit $(p(A))$) :

- si le maître lui dit ($\text{non}(p(A))$), alors l'élève retire $(p(A))$ de sa BC,
- si l'élève croit $(p(X) \rightarrow p_2(X), (p_2(X) \rightarrow p_3(X)), \dots, (p_{n-1}(X) \rightarrow p_n(X))$ et que le maître lui dit ($\text{non}(p_n(A))$) alors un conflit apparaît car l'élève peut déduire $(p_n(A))$ de $(p(A))$ et de la suite d'implication ce qui est contradictoire avec ($\text{non}(p_n(A))$). L'élève devra alors éliminer $(p(A))$ et/ou une ou plusieurs des implications précédentes. Il effectuera alors une série de demandes d'informations au maître,

Nous nommerons "type 2" les conflits où l'élève possède une suite d'implications complétée par le maître qui fournit une implication génératrice de conflit, et "type 1" les autres conflits.

3.4.2 Les élèves curieux

Nos agents peuvent être curieux, comme les sont parfois les agents naturels, et désirer approfondir leur apprentissage au-delà de la leçon du maître.

Élargir la base d'un prédicat : A chaque fois que l'élève intègre une donnée il a la possibilité de s'en servir pour générer de nouvelles connaissances. Ainsi s'il apprend une formule du type $(p(a))$ dont le prédicat est nouveau mais la constante déjà connue dans une autre formule du type $(q(a))$ alors il peut se demander s'il y a un rapport entre les prédicats p et q . Il peut alors questionner le maître sur la validité de $(q(x) \rightarrow p(x))$.

C'est une forme d'induction car, à partir de faits qui possèdent un point commun (la constante), l'élève se pose la question de l'existence d'une règle les reliant.. Si la formule est vraie alors l'élève pourra déduire que toutes les constantes vérifiant q vérifient maintenant aussi p , c'est-à-dire augmenter la base du prédicat p .

Exemple :

; l'élève connaît déjà ($\text{humain}(\text{Baptiste})$), ($\text{humain}(\text{Fabien})$) et ($\text{humain}(\text{Didier})$)

>Maître - *donne* - *connaissance*($\text{animal}(\text{Baptiste})$)

; l'élève rajoute cette connaissance à sa BC et se demande alors si les êtres humains ne seraient pas aussi des animaux

>Élève - *demande* - *information*($\text{humain}(x) \rightarrow \text{animal}(x)$)

>Maître - *donne* - *information*(*Vrai*)

; l'élève peut alors augmenter sa base de ($\text{animal}(\text{Fabien})$) et ($\text{animal}(\text{Didier})$) en plus de $\text{animal}(\text{Baptiste})$ qu'il vient d'apprendre.

Augmenter la connexité de la BC : A tout moment l'élève peut se demander s'il n'existe pas un lien direct entre deux données de sa base qui sont soit reliées par un long chemin soit sans lien (elles appartiennent à deux composantes différentes de la BC).

Nous prenons l'exemple de prédicats qu'il apprend. Ainsi si le maître lui enseigne les deux formules suivantes : ($\text{humain}(x) \rightarrow \text{mortel}(x)$) et ($\text{humain}(x) \rightarrow \text{animal}(x)$) alors l'élève pourra s'interroger sur une éventuelle relation directe entre ($\text{mortel}(x)$) et ($\text{animal}(x)$). Le dialogue lui servira alors à demander au maître s'il a connaissance d'une telle relation.

L'élève peut alors procéder à deux questions :

- *demande* - *information*($\text{mortel}(x) \rightarrow \text{animal}(x)$)

- *demande* - *information*($\text{animal}(x) \rightarrow \text{mortel}(x)$)

Le maître peut, par exemple, lui confirmer la seconde relation. La BC de l'élève s'est donc enrichie d'une nouvelle relation.

Nous pouvons aussi observer un accroissement de la connexité lorsque l'élève tente de réaliser le premier but d'apprentissage, c'est-à-dire lors de l'apprentissage de données nouvelles : Dans l'exemple où l'élève doit connaître le prédicat (*humain(x)*) pour apprendre la nouvelle formule, l'élève connaissait déjà les trois prédicats (*animal(x)*), (*intelligent(x)*) et (*parle(x)*), et en apprenant le nouveau prédicat, les trois anciens se sont vu rajouter un lien entre eux. La connexité s'est alors accrue.

Lors de la gestion de conflits, une implication peut être remise en cause puis éliminée, entraînant une réduction du niveau de connexité de la BC si cette dernière se situait au milieu d'une chaîne d'implications et constituait l'unique point de connexité entre ces deux dernières. Afin de minimiser les chances d'une perte de connexité, l'élève peut utiliser une heuristique de choix d'implication à valider auprès du maître : en commençant par les implications en bout de chaîne (la première et la dernière), si l'une d'entre elles s'avérait à être fausse, alors sa suppression ne changerait pas le niveau de connexité de la base.

Apprendre de nouvelles constantes : Apprendre des prédicats reste théorique, en revanche apprendre les constantes qui valident ces prédicats permet à l'agent de pouvoir appliquer ses connaissances au monde dans lequel il évolue. Ainsi un agent élève curieux pourra, suite à l'apprentissage de nouveaux prédicats, demander au maître s'il connaît des constantes rendant à vrai ces prédicats.

Par exemple si l'élève apprend qu'il ne faut pas s'approcher des agents dangereux (*dangereux(x) → resterloin(x)*), alors il pourrait lui être utile de connaître quelques uns de ces agents. Par exemple (*dangereux(didier)*), (*dangereux(fabien)*), ...

Par ailleurs les valeurs associées à certaines constantes peuvent être utiles à connaître. Par exemple si l'agent apprend que la valeur de la pesanteur sur Terre est g alors il peut y avoir un intérêt de connaître la valeur de cette constante. Ainsi en procédant à une demande de précision concernant cette valeur, l'élève pourra, avec les capacités de calcul nécessaires, l'utiliser pour résoudre des problèmes de gravité.

Comprendre les implications : Après l'apprentissage de toute implication ($P \rightarrow Q$), l'élève peut se demander si (P) implique directement (Q) ou si c'est le résultat d'une série d'implications. C'est le cas typique d'une demande d'explication portant sur une implication. En réalisant une telle demande au maître, l'élève a l'opportunité d'accroître la quantité de données de sa BC tout en augmentant sa connexité.

Chapitre 4

Architecture et conception

L'approche théorique du chapitre 3 a été spécifiée et partiellement mise en oeuvre (en fonction du temps imparti au stage de DEA). La spécification est générale, la conception contient certains éléments de celle-ci que nous préciserons dans la partie 4.2.

4.1 Architecture

La figure 4.1 expose les principaux éléments d'architecture de notre système d'apprentissage par dialogue. Notre système se décline en cinq composants : l'agent maître, l'agent élève, les rôles fonctionnels, les stratégies et le monde.

Chaque agent dispose d'une BC, d'un modèle de "soi" Σ et de son interlocuteur Φ . Il a un accès total à ces trois éléments en lecture et écriture afin de les faire évoluer.

Pour le maître, Σ comprend :

- son rôle de maître,
- son but : enseigner à l'élève, c'est-à-dire transmettre des connaissances qu'il croit vraies.

Φ comprend :

- l'identité de l'élève,
- la nature de l'élève qui consiste à apprendre la leçon enseignée,
- un modèle de la BC de l'élève.

Une partie de la BC du maître constitue la leçon.

Symétriquement, pour l'élève Σ comprend :

- son rôle d'élève,
- son but : apprendre une leçon enseignée par la maître.

Φ comprend :

- l'identité du maître,
- la nature du maître qui consiste à enseigner une leçon,
- la valeur sûre des connaissances fournies par le maître.

Le modèle de la BC du maître est facultatif pour l'élève car il se soucie peu de l'évolution des connaissances du maître.

Ces deux agents disposent de stratégies pour apprendre et enseigner une leçon qui leur servent de méthode pour procéder au cours.

Ils disposent aussi de l'ensemble des rôles fonctionnels (RF) décrits en section 3.3 qui leur permettent de :

- connaître les RF à leur disposition pour communiquer
- savoir interpréter correctement un message d'autrui accompagné d'un certain RF
- échanger les messages

Monde

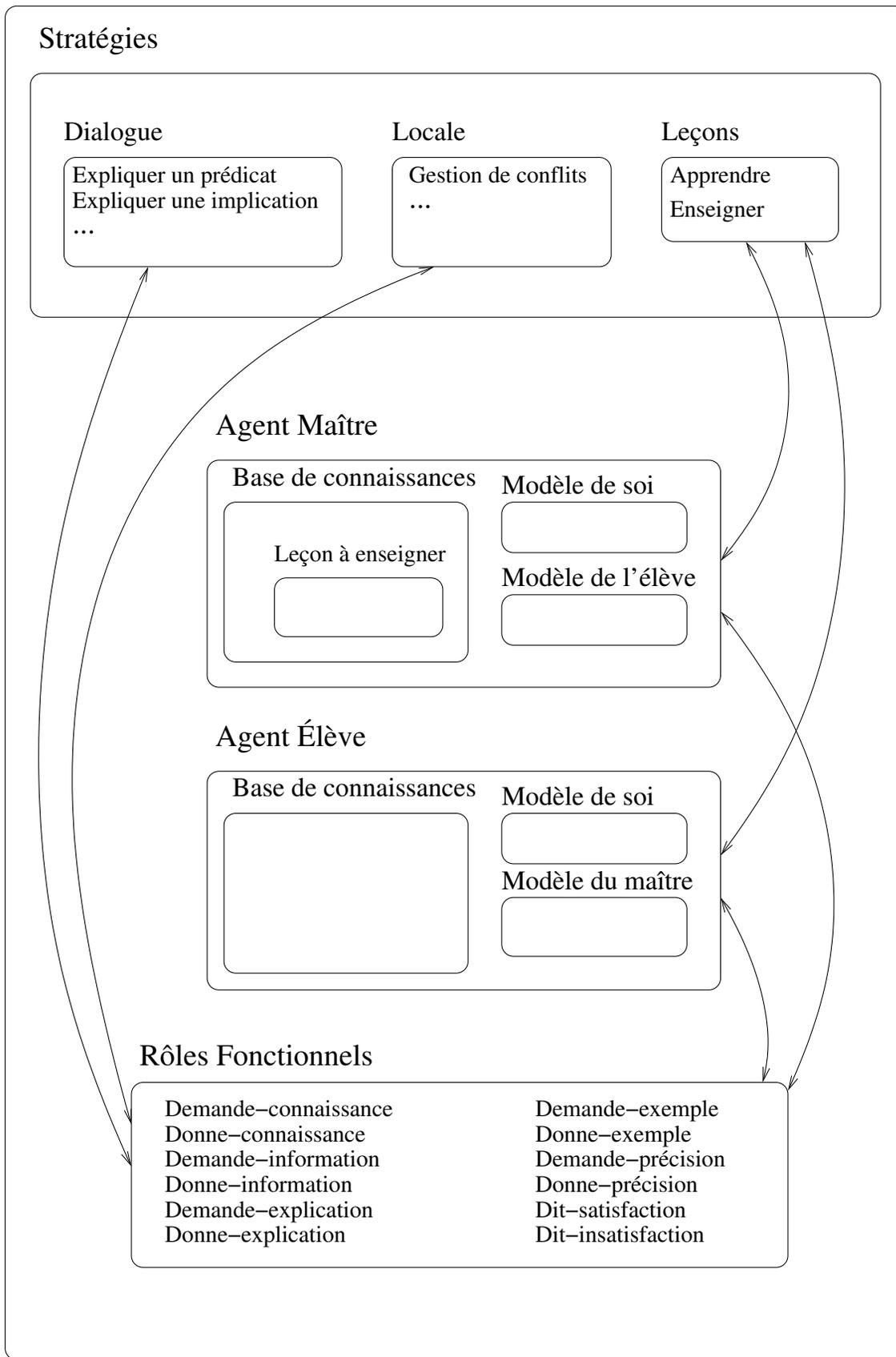


FIG. 4.1 – Le système d'apprentissage par le dialogue entre les agents cognitifs

Le but du composant des RF est d'expliquer aux agents comment procéder face à un message provenant d'un autre agent. Pour ce faire, les RF ont accès aux stratégies de dialogue et aux stratégies locales.

Par exemple en recevant une demande d'explication d'un prédicat, le maître va la soumettre au composant des RF afin que la bonne stratégie soit utilisée pour répondre au mieux à l'élève. Actuellement ce sera la stratégie de dialogue d'explication de prédicat qui sera utilisée.

Certaines stratégies locales seront parfois utilisées comme la gestion de conflit dans le cas d'une transmission de connaissance du maître vers l'élève (RF donne-connaissance) qui entre en contradiction avec la BC de l'élève.

Les stratégies sont actuellement établies manuellement et accessibles en lecture seule par les agents et le composant RF. Cependant, si les agents disposaient d'une marge de main d'oeuvre pour accomplir les stratégies et d'un système d'apprentissage de stratégies, ils pourraient être en mesure de créer leurs propres stratégies à partir de leurs expériences.

4.2 Conception

Afin de mettre en pratique une partie de notre théorie, nous avons réalisé un programme en Java. Les capacités de raisonnement de nos agents nous ont conduits à utiliser un langage de programmation logique, nous avons choisi Prolog. Il existe plusieurs intégrations de Prolog en Java comme tuProlog, JIP, DGKS Prolog, JLog ou JavaLog. Nous avons opté pour tuProlog pour sa simplicité et son efficacité, sans contrainte d'enregistrement du produit.

Prolog nous restreint à des formules de type clause de Horn. Les données connues et échangées sont donc simplifiées mais la contrainte que nous nous sommes fixés est conservée : c'est toujours une situation naturelle de dialogue.

Prolog utilise un programme qui contient les règles nécessaires au raisonnement logique et des questions afin d'extraire les informations requises. Nous utilisons la syntaxe suivante [6] :

```

<programme> ::= <règle>*
<règle>      ::= <fait>.
              ::= <tête> :- <queue>.
<fait>      ::= <atome>
<tête>      ::= <atome>
<queue>     ::= <atome> [ , <atome> ]*
<question> ::= <atome> [ , <atome> ]*

```

- un atome est, comme en logique du 1^{er} ordre, de la forme :
 - p où p est un symbole de prédicat d'arité 0,
 - ou $p(t_1, t_2, \dots, t_n)$ où p est un symbole de prédicat d'arité n et les t_i sont des termes (variables, constantes ou termes construits à partir de symboles de fonction),
- les symboles de variables commencent par une majuscule,
- les symboles de prédicats, de fonctions et de constantes, commencent par une minuscule,

Une implication logique $P \rightarrow Q$ sera notée en prolog "Q :- P." et un fait P sera noté "P.". Nous utiliserons le prédicat de prédicats `not(P)` comme négation logique.

Nous avons créé un environnement contenant deux agents : un maître et un élève. Chaque agent a pour BC un programme et s'il souhaite extraire des informations de sa BC il peut soit utiliser une question prolog, soit une des méthode spécifique d'accès que nous avons développée.

La structure de notre programme est celle présentée dans la section 4.1. Nous avons implémenté les rôles fonctionnels, les stratégies de leçon, et la stratégie de gestion de conflits.

Nous avons pourvu nos agents :

- de connaissances de base, avec une bonne partie de celles de l'élève incluses dans celles du maître,
- de données à enseigner, uniquement pour le maître,
- de la faculté d'apprentissage de nouvelles données,
- d'accès aux stratégies de leçon,
- d'aptitudes à dialoguer avec un autre agent,
- de méthodes d'accès aux rôles fonctionnels afin d'interpréter correctement les messages entrants et d'utiliser les bonnes stratégies.

Nous avons choisi d'implémenter la gestion des conflits d'implication de type 1 car c'est une situation qui peut arriver fréquemment, pour des agents artificiels comme naturels, et qui nécessite un traitement délicat afin de conserver une BC cohérente et d'éliminer le moins d'informations possibles.

4.3 Les classes java

Le programme java est découpé en cinq classes :

- la classe World qui représente le monde, l'environnement dans lequel évoluent nos agents. Ces derniers sont donc des attributs de cette classe. Tout message passe par cette classe avant d'être transmis à un autre agent,
- la classe AgentCog qui représente un agent cognitif, maître comme élève car toute agent est potentiellement l'un ou l'autre. Elle dérive de la classe Threads afin de faire coexister simultanément tous les agents,
- la classe VectTheory qui contient le programme prolog — autrement appelé théorie —, c'est donc la BC. Elle dérive de la classe Vector car c'est une simple liste d'implications et de faits. Chaque agent possède une et une seule théorie,
- la classe RoleFctl qui contient l'ensemble des RF avec pour chacun une méthode de traitement d'un message,
- la classe Stratégies qui regroupe les stratégies de leçon, de dialogue et les stratégies locales utilisées par les classes AgentCog et RoleFctl.

4.4 Les algorithmes de gestion du conflit

L'agent élève, lorsqu'il reçoit une nouvelle donnée du maître, doit avant de l'ajouter à sa BC tester si cette dernière n'est pas contradictoire avec sa BC. Nous traiterons du cas de l'ajout d'une implication $p(x) \rightarrow q(x)$, c'est le type 1 des conflits d'implications. Pour exécuter ce test il recherche dans sa BC s'il dispose d'une implication $p(x) \rightarrow non(q(x))$.

L'algorithme de recherche est le suivant :

CHERCHE CONTRADICTION

entrée : $p(x)$, $non(q(x))$

sortie : liste d'implication $p(x) \rightarrow \dots \rightarrow non(q(x))$ si elle existe,
liste vide sinon.

$L <-$ liste vide d'implication ;

CHERCHE CONTRADICTION RECURSIF($p(x)$, $non(q(x))$, L) ;

retourner L

CHERCHE CONTRADICTION RECURSIF

entrée : $r(x)$, $s(x)$, L

sortie : *Vrai* si la première prémisses de L est $r(x)$

et si la dernière conclusion de L est $s(x)$,

Faux sinon.

Pour chaque implication $I = t(x) \rightarrow u(x)$ de la BC faire

Si $r(x) = t(x)$ et $u(x) \notin L$

 Si $u(x) = s(x)$

$L <- L + t(x) \rightarrow u(x)$

 retourner *Vrai*

$L <- L + t(x) \rightarrow u(x)$

 CHERCHE CONTRADICTION RECURSIF($u(x)$, $s(x)$, L)

Si L non vide alors retirer dernière implication de L

retourner *Faux*

Le test $u(x) \notin L$ revient à vérifier si $u(x)$ est égal à aucune des conclusions des implications de L .

Si l'algorithme retourne la liste vide alors il n'y a pas de contradiction. Dans le cas contraire, au moins une des implications de la liste L est fautive car contradictoire avec l'implication fournie par le maître. L'élève va alors demander au maître de valider chaque implication puis éliminera les fautes :

ELIMINE CONTRADICTION

entrée : L

sortie : rien, en revanche la BC sera modifiée

Pour chaque implication $I = p(x) \rightarrow q(x)$ de L faire

Envoyer au maître *demande – information(I)*

Si je reçois du maître *donne – information(Faux)* alors

Supprimer I de la BC

Sinon ne rien faire

Une fois ce travail effectué, l'élève peut enfin ajouter à sa BC l'implication fournie par la maître.

Chapitre 5

Mise en oeuvre et évaluation

Afin de tester la réalisation de notre théorie, nous avons effectué plusieurs exécutions de notre programme avec, pour chacune, des BC et leçons différentes.

5.1 Premier exemple : un conflit d'implications

Les connaissances de base ($\Gamma(\textit{agent})$) de nos deux agents sont :

Pour le maître ($\Gamma(M)$) :

```
mortel(x) :- humain(x).
vivant(x) :- mortel(x).
vivant(x) :- tuable(x).
reproduit(x) :- vivant(x).
```

Pour l'élève ($\Gamma(E)$) :

```
tuable(x) :- humain(x).
vivant(x) :- tuable(x).
not(reproduit(x)) :- vivant(x).
```

Les données à enseigner du maître à l'élève sont ($\delta(M)$) :

```
mortel(x) :- humain(x).
vivant(x) :- humain(x).
reproduit(x) :- humain(x).
```

Nous avons naturellement imposé la contrainte :

$$\delta(M) \subset \Gamma(M)$$

Le maître ne peut enseigner que des données qu'il connaît déjà. Nous considérons toutefois que $\Gamma(M)$ possède toutes les implications déduites de celles déjà présentes.

Par exemple :

$$(\textit{humain}(x) \rightarrow \textit{vivant}(x)) \in \Gamma(M)$$

Le maître va ensuite enseigner chaque formule à l'élève, en attendant entre chaque une satisfaction ou des questions, voici le dialogue :

>Maître - *donne – connaissance*($humain(x) \rightarrow mortel(x)$)
; cette formule est apprise sans difficultés par l'élève.
>Élève - *dit – satisfaction*
>Maître - *donne – connaissance*($humain(x) \rightarrow vivant(x)$)
; l'élève connaît déjà cette formule (déduite de sa BC). Il ne l'apprend pas mais la comprend.
>Élève - *dit – satisfaction*
>Maître - *donne – connaissance*($humain(x) \rightarrow reproduit(x)$)
; cette nouvelle donnée entre en contradiction avec $\Gamma(E)$, l'élève va devoir localiser sa ou ses connaissances fausses parmi celles qui génèrent le problème, c'est-à-dire parmi $humain(x) \rightarrow tuable(x)$, $tuable(x) \rightarrow vivant(x)$ et $vivant(x) \rightarrow not(reproduit(x))$
>Élève - *demande – information*($humain(x) \rightarrow tuable(x)$)
>Maître - *donne – information*(*Inconnu*)
; le maître ne sait pas si un être humain peut être tué, l'élève conserve toutefois cette formule.
>Élève - *demande – information*($tuable(x) \rightarrow vivant(x)$)
>Maître - *donne – information*(*Vrai*)
>Élève - *demande – information*($vivant(x) \rightarrow not(reproduit(x))$)
>Maître - *donne – information*(*Faux*)
; cette donnée entre en contradiction avec la base du maître donc il la juge fausse.
; l'élève supprime donc cette donnée de sa BC.
; son conflit géré il peut maintenant apprendre la connaissance ($humain(x) \rightarrow reproduit(x)$) en toute sécurité.
>Élève - *dit – satisfaction*
; il termine en informant sa satisfaction au maître

Voilà la nouvelle BC de l'élève :

```

tuable(x) :- humain(x).
vivant(x) :- tuable(x).
animal(x) :- humain(x).
mortel(x) :- humain(x).
reproduit(x) :- humain(x).

```

5.2 Deuxième exemple : un conflit de faits

Les connaissances de base ($\Gamma(agent)$) de nos deux agents sont maintenant :

Pour le maître ($\Gamma(M)$) :

```

humain(jean).
mortel(x) :- humain(x).
vivant(x) :- mortel(x).
vivant(x) :- tuable(x).
reproduit(x) :- vivant(x).

```

Pour l'élève ($\Gamma(E)$) :

```

humain(jean).
tuable(x) :- humain(x).
vivant(x) :- tuable(x).
not(reproduit(x)) :- vivant(x).

```

Les données à enseigner du maître à l'élève sont $(\delta(M))$:

`reproduit(jean)`.

Nous notons que la situation est proche de la précédente car elle doit amener l'élève à faire les mêmes conclusions sur sa connaissance ($vivant(x) \rightarrow not(reproduit(x))$).

Voici le dialogue :

>Maître - *donne – connaissance*(*reproduit(jean)*)
; cette nouvelle donnée entre en contradiction avec $\Gamma(E)$, 4 données vont alors être testées, comme nous l'avons vu dans la section 3.4
>Élève - *demande – information*(*humain(jean)*)
>Maître - *donne – information*(*Vrai*)
>Élève - *demande – information*(*humain(x) → tuable(x)*)
>Maître - *donne – information*(*Inconnu*)
>Élève - *demande – information*(*tuable(x) → vivant(x)*)
>Maître - *donne – information*(*Vrai*)
>Élève - *demande – information*(*vivant(x) → not(reproduit(x))*)
>Maître - *donne – information*(*non*)
>Élève - *dit – satisfaction*

Voilà la nouvelle BC de l'élève :

`humain(jean)`.
`tuable(x) :- humain(x)`.
`vivant(x) :- tuable(x)`.
`reproduit(jean)`.

À travers ces deux exécutions nous pouvons observer que l'agent élève détecte lorsqu'un conflit apparaît entre sa BC et une donnée fournie par le maître, puis demande au maître de valider certaines connaissances potentiellement conflictuelles pour enfin éliminer celles dont le maître sait qu'elles sont fausses.

Tous ces conflits ont donc été gérés correctement, en supprimant le moins d'informations possibles.

Chapitre 6

Conclusion et perspectives

Ce mémoire présente une modélisation et une réalisation informatique de l'apprentissage par le dialogue entre agents artificiels cognitifs, dans une relation de type maître — élève. Comme on peut en témoigner sur le plan humain, le dialogue est un mode de communication très adapté à l'apprentissage. C'est pourquoi nous avons souhaité en faire bénéficier les agents artificiels. Nous avons pourvu ces agents d'une base de connaissance (BC) dans le langage de la logique des prédicats afin qu'ils puissent raisonner. La notion de *connexité* de la BC a été définie et nous a permis de mieux évaluer le niveau de relation entre les données de l'élève. La volonté d'augmenter la connexité de la BC est alors apparue comme un but à part entière, visant à mieux relier toute connaissance connue. L'utilisation de la théorie des rôles fonctionnels a grandement facilité la modélisation des dialogues, permettant à chaque agent d'interpréter correctement le type de message qu'il reçoit.

Nous avons ensuite étudié les solutions à des difficultés d'incompréhension de l'élève face à des données fournies par le maître. Pour les problèmes de prédicats inconnus nous avons présenté une stratégie de dialogue inspirée de la méthode socratique. Pour la gestion des données contradictoires plusieurs cas de conflits ont été distingués et des algorithmes adéquats ont été proposés.

Enfin nous avons traité du cas des élèves curieux, c'est-à-dire désireux d'apprendre plus que la leçon fournie par le maître. Ces élèves tenteraient alors d'élargir la base de leurs prédicats, d'augmenter la connexité de leur BC, d'apprendre de nouvelles constantes et de comprendre les raisons d'être de certaines implications.

Le programme java que nous avons réalisé nous a permis de tester une partie de notre théorie. Un agent maître et un agent élève ont été créés dans un environnement permettant le dialogue. Quelques connaissances de base et une leçon à enseigner (pour le maître) leur ont été fournies au préalable. Ces agents avaient à leur disposition des stratégies de leçon et une stratégie locale de gestion de conflits, l'ensemble des rôles fonctionnels et les facultés nécessaires de dialogue et d'apprentissage. Nous avons testé la gestion des conflits de données dans deux situations : la première générant un conflit d'implications, la seconde générant un conflit de faits. Dans les deux cas l'agent élève a su détecter l'incohérence et par le dialogue localiser là où sont les sources de contradiction afin de les éliminer.

Ce travail ne constitue qu'une première approche à l'apprentissage par le dialogue entre agents artificiels cognitifs. De nombreuses améliorations sont réalisables comme la création de nouvelles stratégies de dialogue afin de mieux expliquer les connaissances à l'élève, de stratégies de leçon afin d'optimiser la structure générale du cours, ou de stratégies locales afin d'améliorer des parties spécifiques comme la gestion des conflits.

L'implémentation voit aussi des évolutions possibles comme une généralisation du type des données afin qu'elles supportent les conjonctions de prédicats ou les équivalences entre formules.

La gestion des élèves curieux pourrait grandement améliorer l'apprentissage. Des expériences à plus grande échelle pourraient être réalisées mettant en jeu plus de deux agents dont le rôle de chacun varierait de maître à élève. Ainsi, l'évolution de l'ensemble des connaissances pourrait être observé et analysé.

Bibliographie

- [1] J.Allen et R.Perrault,
Analyzing intention in utterances,
Artificial Intelligence, 15(3) :143-178, 1980.

- [2] Diego Andina de la Fuente, M. en I. Antonio Vega Corona,
ARTIFICIAL NEURAL NETWORKS,
URL : “<http://www.gc.ssr.upm.es/inves/neural/ann1/anntutorial.html>”.

- [3] J.L. Austin,
How to Do Things with Words,
second edition, ed. J. O. Urmson and Marina Sbisa,
Cambridge, Mass : Harvard University Press, 1975.

- [4] P.Cohen et H.Levesque,
Intentions in Communication, chap. 12,
Rational Interaction as the Basis for Communication. Bradford Books, MIT Press, seconde édition,
1992.

- [5] Jacques Ferber,
LES SYSTÈMES MULTI-AGENTS, Vers une intelligence collective, InterEditions, 1995.

- [6] Philippe Janssen,
Cours de prolog, Maîtrise d’informatique, 26 février 2002.

- [7] C.Kerbrat-Orecchioni,
L’énonciation, de la subjectivité dans le langage, 1970.

- [8] Thierry Lemeunier,
L’intentionnalité communicative dans le dialogue homme-machine en langue naturelle,
Thèse, pages 17-37, 2000

- [9] D. Luzzati,
Le dialogue verbal homme-machine, Masson, Paris, 1995.

- [10] Leslie Pack Kaelbling,
Reinforcement Learning : A Survey, Journal of Artificial Intelligence Research 4, pages 237-285, 1996.

- [11] Violaine Prince,
Dialogue et Web : La théorie des rôles fonctionnels,
URL : “[http://www.lirmm.fr/~prince/VP-enseignement/DEA/Dialogue et Web/rolesfonct.ppt](http://www.lirmm.fr/~prince/VP-enseignement/DEA/Dialogue%20et%20Web/rolesfonct.ppt)”.

- [12] Violaine Prince,
Raisonnements dans les systèmes logiques formels et leurs représentativité des systèmes cognitifs naturels
Rapport Interne de LRIA Université Paris 8, Deuxième édition, 1999, ISSN 01286-7993

- [13] Vikram Pudi,
Neural Networks,
URL : “http://dsl.serc.iisc.ernet.in/~vikram/nn_intro.html”.
- [14] Searle,
Speech Acts : An Essay in the Philosophy of Language,
Cambridge : Cambridge University Press, 1969.
- [15] Christos Stergiou and Dimitrios Siganos,
NEURAL NETWORKS,
URL : “http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html”.
- [16] Peter W. Eklund,
A Performance Survey of Public Domain Supervised, Machine Learning Algorithms,
School of Information Technology and Electrical Engineering, The University of Queensland St.,
Lucia Queensland 4072, Australia.

MODÉLISATION DE L'ACQUISITION DE CONNAISSANCES PAR LE DIALOGUE ENTRE
AGENTS COGNITIFS

MEHDI YOUSFI MONOD

résumé :

Ce travail se situe dans le cadre de la communication et l'apprentissage des agents artificiels cognitifs. Nous nous concentrons sur le dialogue comme unique moyen d'apprentissage de nos agents, car il est trop peu utilisé et étudié en recherche informatique. Ce mémoire vise à fournir une modélisation d'un dialogue permettant à deux agents, dans une relation maître — élève, d'échanger des informations dans le but d'apprendre. Nous définissons ensuite la nature de nos agents, les types de relations qu'ils entretiennent et la structure et le contenu de leur base de connaissances. Nous continuons en expliquant comment nous avons introduit la théorie des rôles fonctionnels pour modéliser les échanges dialogiques. Nous traitons par la suite des différents but d'apprentissage, de leur réalisation et des solutions aux difficultés rencontrées par les agents. Une architecture générale est alors établie et une implémentation d'une partie de la théorie est réalisée. Nous concluons sur les réalisations effectuées et les améliorations potentielles.

mots clés : dialogue machine-machine, apprentissage machine, systèmes multi-agents, raisonnement logique

abstract :

This work takes place in the framework of cognitive artificial agents learning and communicating. We concentrate on dialogue as the unique way for acquiring knowledge for our agents, since this has not really been used and studied in computer science research. This dissertation aims at providing a modelling of a dialogue allowing two agents, in a master — student relationship, to exchange information with a motivation of learning. Then we define the nature of our agents, the types of relation which they maintain, and the structure and contents of their knowledge base. We continue by explaining how we introduced the functional roles theory to model dialogical exchanges. Then we talk about the different aims of learning, their realization and the solutions we provide for problems encountered by our agents. A general architecture is then established and an implementation of a part of the theory is carried out. We conclude about the achievements carried out and the potential improvement of our work.

Keywords : machine-machine dialogue, machine learning, multi-agents systems, logical reasoning

Directeur de stage : Violaine Prince