

BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS

COMPUTATION - INFORMATION

N° 77 - JUIN 2007

DIRECTEUR

Jean - Michel Knippel

FONDATEUR

Edmond Bianco

SERVEUR DE PUBLICATION

Christian Blanvillain

SECRETARIAT

Kalassoumi Adjilani

Université de Provence
Equipe Hermès. Case 33
3 place Victor Hugo
F - 13331 Marseille Cedex 3
Téléphone: (0)4 91 10 62 30
Télécopie : (0)4 91 50 91 10

DEPOSITAIRE

Université de Provence
Bibliothèque Universitaire
1 place Victor Hugo
F - 13331 Marseille Cedex 3
Téléphone : (0)4 91 10 85 29
Télécopie : (0)4 91 95 75 57

IMPRIMEUR

Université de Provence
Service Reprographie
3 place Victor Hugo
F - 13331 Marseille Cedex 3
Téléphone : (0)4 91 10 60 48

1 EDITORIAL La 9

par Jean - Michel Knippel

3 RespiDiag : un système Multi-Agents Basé sur un Couplage CBR-SBC pour la Prise en Charge de la Santé Respiratoire

par Souad Guessoum, Mohamed Tayeb Laskri, Rachid Benali

9 Un Système Tuteur Utilisant un Compagnon Perturbateur

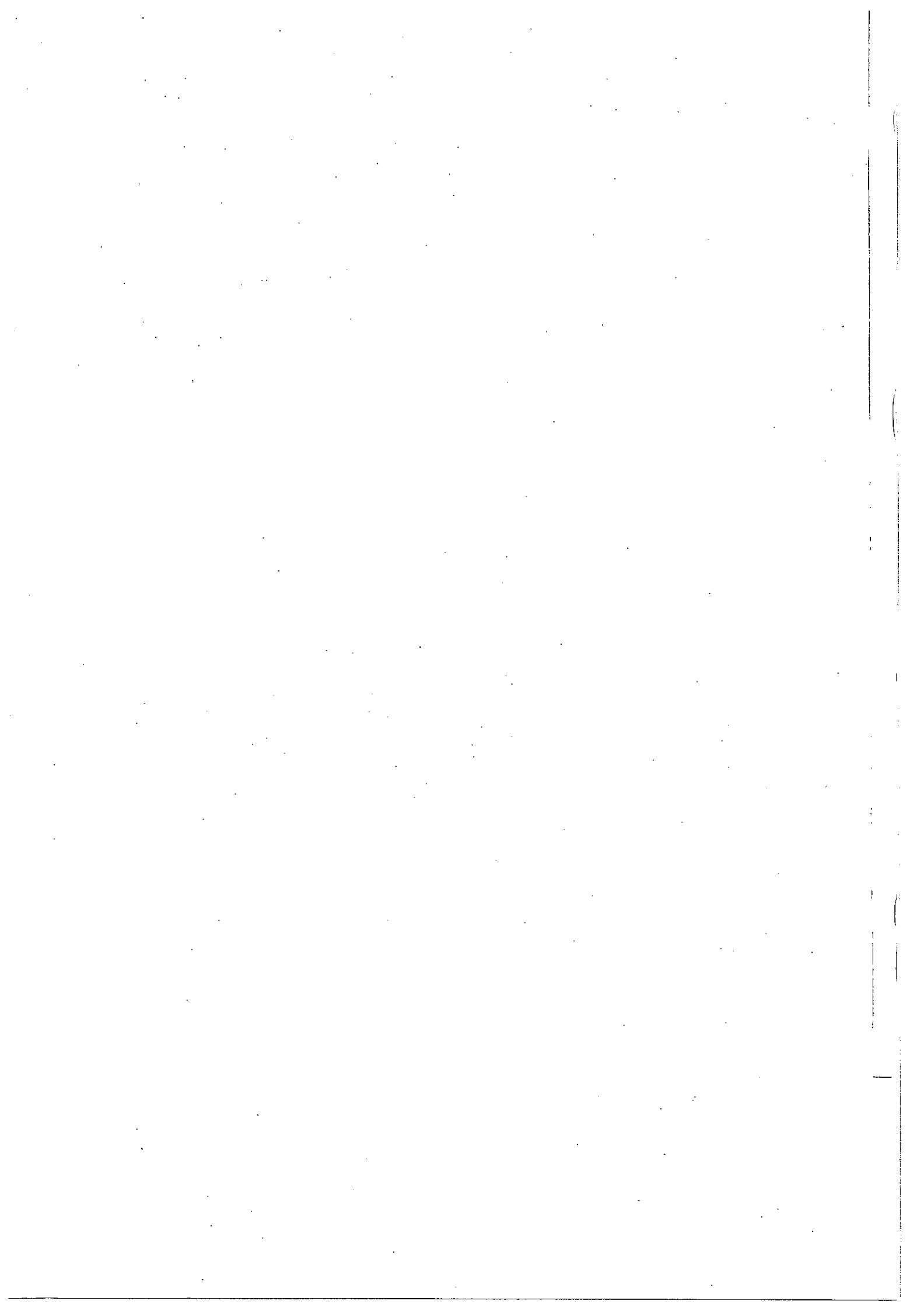
par Tahar Bouhadada, Mohamed Tayeb Laskri

17 VOZZAVEDIBISAR Fantaisie

par Edmond Bianco

<http://www.univ-provence.fr/biaa>

Publication trimestrielle, gratuite, de l'Université de Provence



EDITORIAL

La 9

Jean - Michel Knippel

Vous pourriez penser que je vais vous parler de téléphonie ou d'une chaîne de télévision, et bien non. Vous avez sûrement d'autres évocations sur ce titre. C'est le langage des conducteurs de bus : « la 9 ».

Je ne parlerai que d'une ligne de bus qui passe par ma « ville » de Roquevaire où coule le fleuve qui se jette dans la mer à Marseille. Marseille est à côté de Roquevaire depuis que le clavier personnel de Pierre Cochereau est arrivé dans l'église de Roquevaire. Avant, Roquevaire se trouvait à côté de Marseille. Le festival d'orgue international est bien présent et grandit. Naturellement, certains hauts personnages de notre France, qui rigolaient de cette initiative, sont assis au premier rang des concerts, que du normal. Pourquoi ? C'est le « même » clavier que celui de l'église Notre Dame de Paris.

INFORMATION

Un matin de printemps, je me décidais à prendre le bus pour Aubagne, puis le train pour Marseille, histoire de laisser motocyclette et automobile au garage. La gare de Roquevaire existe, elle est évoquée au moment des élections : « il faut la réouvrir, les routes sont saturées ». Ce sera pour la prochaine, elle se repose à l'ombre des platanes depuis des dizaines d'années et la route va voir de plus en plus de voitures avec de nouvelles constructions voisines de la voie ferrée. Etonnant ?

COMMUNICATION

Quelques piétons attendaient un bus du matin, avec moi. Après le quart d'heure de retard possible, tout le monde se renseigne. Des personnes passent en voiture et nous disent que la route est coupée par un accident grave. Elle est coupée, mais seules les voitures circulent ! Coup de téléphone à la gare routière, « pas de problème, les bus roulent ». Mais nous sommes dans le village de Roquevaire, au pont sur l'Huveaune. Toujours aucun problème pour la compagnie de bus. Naturellement, les bus sont équipés de moyens de communication. A quoi ont-ils servi dans ce cas-là ? Par où passaient les bus de « la 9 » ? Les bus roulaient sur l'autoroute en « banlieue » de Roquevaire où ne coule pas l'Huveaune pour contourner l'accident. Que je sache, les bus n'ont pas de station sur les autoroutes.

COMPUTATION

Tout calcul fait, où il faut une heure tout compris pour aller de Roquevaire à Marseille, l'expédition a pris cinq heures, le temps que les bus reprennent la voie normale. Il n'y avait pas d'ecclésiastiques pour faire cette supputation du temps, mais une meilleure communication avec les outils que nous avons aurait fait couler le temps d'une autre manière.

Rappelons ici un sens premier du mot computation qui a donné « ordinateur » supplanté par « ordinateur ». Ce terme est issu du latin « computatio » (calcul). C'est une méthode utilisée pour calculer une date ou une durée. Le comput, emprunté au latin « computus » (compte, calcul), est un ensemble de calculs visant à déterminer la date de Pâques et des autres fêtes mobiles. Ces calculs sont sujets à des corrections dues aux mouvements « irréguliers » de rotation de la lune qui entre en ligne de compte dans la méthode de calcul. La rotation des bus de la ligne « 9 » a donc aussi ses irrégularités. J'oublie la possibilité de prise en compte de l'humeur de notre fleuve : l'Huveaune et d'autres phénomènes de notre bon vieux système solaire qui perturbent de temps en temps les horaires.

RespiDiag : Un Système Multi-Agents Basé sur un Couplage CBR-SBC pour la Prise en Charge de la Santé Respiratoire

GUESSOUM Souad⁽¹⁾ - LASKRI Mohamed Tayeb⁽²⁾ – BENALI Rachid⁽³⁾
⁽¹⁾Département d'Informatique, ⁽²⁾Département de Médecine
Université de Badji Mokhtar Annaba, B.P. 12 - Algérie
souadguessoum@yahoo.fr, laskri@univ-annaba.org, ben_rk@yahoo.fr

Résumé

La médecine constitue un excellent champ d'expérimentation pour l'intelligence artificielle (IA). Dans ce travail, une modélisation basée agents et combinant plusieurs paradigmes de l'IA est présentée. Elle a pour objectif d'apporter un outil d'aide à la décision pour la prise en charge de la santé respiratoire. Voulant reproduire le raisonnement médical, les règles de production ont été couplées avec le raisonnement à base de cas dans une même entité agent. Et pour permettre un certain degré de parallélisme, la connaissance médicale a été répartie entre plusieurs agents qui travaillent en collaboration pour détecter le diagnostic le plus probable.

Mots-clés — Diagnostic, Aide à la décision, SMA, CBR, SBC

1. Introduction

L'Intelligence Artificielle et la Médecine entretiennent d'étroites relations depuis fort longtemps, chacune ayant justifié l'intérêt qu'elle porte pour l'autre. L'IA tente de découvrir les secrets de l'activité médicale avec tous ses aspects, dans l'objectif de reproduire le raisonnement humain sur machine. D'un autre côté, le domaine de la médecine a toujours constitué un champ d'expérimentation intéressant pour tester les différents paradigmes de l'IA. Ces tests contribuent fortement à l'évolution de ses techniques. Réciproquement, cette évolution est très bénéfique pour la médecine qui en profite largement en obtenant des systèmes informatiques d'aide au diagnostic, d'interprétation d'images radiologiques, de formation et d'enseignement.

Dans notre travail, et encore une fois, l'IA et la médecine se rencontrent. Nous essayons de rassembler des connaissances venues des deux domaines pour réaliser un système informatique d'aide au diagnostic pour la prise en charge de la santé respiratoire des patients.

Le choix de cette spécialité spécifique de la médecine a été motivé par plusieurs facteurs, le plus important étant le fait qu'en Algérie, les symptômes respiratoires représentent le principal motif de consultation dans les services de santé. Le recours des malades vers les services de pneumologie ne cesse d'augmenter. Il est encore mis en évidence par les statistiques des dernières années qui montrent qu'en moyenne, 30% des malades consultent pour des symptômes respiratoires. Chez les enfants de moins de 15 ans, cette proportion s'élève jusqu'à 50%, surtout dans la tranche d'âge (0,4) ans; chez les adultes la proportion est de 20%, et s'élève légèrement chez les plus de 50 ans [Benali et al.04].

Ces symptômes respiratoires sont en rapport avec des infections respiratoires aiguës dans 85% des cas, et des maladies respiratoires chroniques non transmissibles dans 15% des cas environ [Benali et al.04]; principalement l'asthme chez l'enfant, et plus rarement des bronchopathies chroniques : dilatation des bronches chez l'enfant et bronchite chez l'adulte.

En raison de leur fréquence, les maladies respiratoires constituent une part importante de la consommation de médicaments et de soins médicaux (consultations, examens complémentaires). Dans ces circonstances, un outil informatique d'aide à la décision de diagnostic s'avère indispensable pour apporter soutien et appui aussi bien aux médecins généralistes que spécialistes de pneumologie dans leur activité quotidienne.

Dans cette optique, nous nous sommes lancés dans le projet « RespiDiag » qui consiste en une modélisation multi-agents basée sur un couplage de deux modes de raisonnements, à savoir, les règles de production et le raisonnement à base de cas.

La deuxième section de ce papier présente le domaine d'application et les pathologies concernées par le champ d'étude, alors que la troisième est consacrée au système proposé « RespiDiag ». Elle expose son architecture avec ses différents composants, les liens et les interactions entre les agents, les modes de raisonnement de ces derniers ainsi que le

fonctionnement global du système. Après une conclusion de ce travail, quelques perspectives envisagées sont mentionnées.

2. Le Domaine d'application

2.1. Le raisonnement du médecin

Dans ses consultations, le médecin est orienté par les symptômes du patient vers le diagnostic. Généralement, un malade présente plusieurs symptômes dont certains sont plus dominants que d'autres. Le médecin pense alors à tous les diagnostics pouvant causer ces symptômes, puis il procède à une élimination de ceux non pertinents en fonction des symptômes secondaires ou encore en fonction des examens complémentaires (radios, analyses ...) qu'il demandera. *Cette manière de raisonner est principalement basée sur les connaissances académiques du médecin.*

Face à un nouveau malade, le médecin peut également faire appel à ses consultations passées, pour se souvenir des patients ayant présenté des combinaisons de symptômes plus ou moins similaires à ceux du patient actuel. Cela lui rappellera en conséquence, non seulement le diagnostic retenu parmi ceux suspectés, mais aussi les traitements ayant été plus efficaces que d'autres. *Cette manière de raisonner est basée sur les expériences passées du médecin.*

Ainsi, la connaissance médicale est enregistrée sous forme de scripts dans la mémoire du médecin [Schmidt et al.90]. Et lorsque celui-ci reconnaît l'image clinique d'un patient, ces scripts reviennent dans son esprit et il fera un choix du diagnostic ainsi que sa thérapie en identifiant les analogies entre la situation actuelle et celles cryptées [Heinisch et al.99].

2.2. Les pathologies du champs d'étude

Dans notre projet, quatre pathologies ont été étudiées. Elles représentent les maladies respiratoires les plus prévalentes aussi bien à l'échelle nationale qu'à l'échelle mondiale, à savoir la tuberculose, l'asthme, la BPCO et les infections. Leur prévalence est confirmée par les statistiques de l'Organisation Mondiale de Santé (OMS) dont on présente un bref aperçu à la section 5.

a) **La Tuberculose** : C'est une maladie *contagieuse* causée par le *Mycobacterium Tuberculosis* (BK). La voie de transmission est *aérienne* et *interhumaine*, elle se fait principalement par la toux et l'éternuement quand le contact est régulier et étroit.

b) **L'Asthme** : C'est une maladie inflammatoire chronique des voies aériennes faisant intervenir plusieurs cellules en réponse à un allergène. Cliniquement, il se traduit par une dyspnée sifflante

paroxystique expiratoire sur terrain d'atopie. Sur le plan fonctionnel, il se manifeste par un trouble ventilatoire obstructif *réversible* spontanément ou sous broncho-dilatateur. Le traitement de l'Asthme est symptomatique. Il consiste en des prises de broncho-dilatateur qui servent à calmer les crises d'asthme et à diminuer leur fréquence.

c) **La BPCO** : pour Broncho-Pneumopathie Chronique Obstructive est une maladie causée principalement par le tabac. Elle se traduit par une dyspnée sifflante chez les fumeurs âgés. Sur le plan fonctionnel, elle se manifeste par un trouble ventilatoire obstructif *non réversible*.

d) **Les infections respiratoires** : On distingue 2 types : La bronchite et la pneumopathie. La bronchite se manifeste par une toux et une expectoration confirmée par une radio thorax normal. Alors que la pneumopathie se traduit par une toux, une expectoration, des douleurs thoraciques et surtout une fièvre, elle est confirmée par une image pathologique dans la radio thorax et une hyperleucocytose dans la Formule de la Numérotation Sanguine (FNS).

2.3. Quelques statistiques

D'après un rapport publié par l'OMS en 1999, les maladies respiratoires ont représenté en 1998 un total de 15.7% des causes de mortalité dans le monde avec 6.4 pour les affections respiratoires aiguës, 4,2 pour les maladies pulmonaires chroniques, 2.8 pour la tuberculose et 2.3 pour le cancer bronchique. Selon une enquête nationale de santé menée en 1990, les taux des différentes affections respiratoires par 100.000 habitants sont les suivants :

Asthme	800
Pneumonies	611
Bronchite chronique (prévalence)	500
Bronchite chronique obstructive	125
Tuberculose toutes formes	46

Table 1 : Taux des différentes affections respiratoires/100000h

En 1997, le monde compte [Nunn.00] 7.95 millions de nouveaux cas de tuberculose, une prévalence de 16.2 millions, 1.87 millions de décès. Notons que l'Algérie compte 18.000 tuberculeux pour l'année 2006.

3. Le Système RespiDiag

3.1. Introduction

La fiabilité d'un système informatique médical est proportionnelle à la distance entre le principe des paradigmes informatiques utilisés et la nature du raisonnement médical. Voulant maximiser la performance de notre système, nous avons étudié différents concepts et paradigmes de l'IA pour nous rapprocher au mieux du raisonnement médical.

En observant le raisonnement du médecin sur ses connaissances académiques, il s'est avéré que les règles de production le reproduisent parfaitement. Et en cherchant le raisonnement qui modélise le mieux le recours du médecin à ses expériences passées, le choix s'est porté sur le raisonnement à base de cas.

Dans cette optique, un couplage entre les règles de production et le raisonnement à base de cas nous a semblé assez prometteur. Et dans l'objectif d'augmenter l'efficacité et les performances du système, le concept des agents a été introduit. La connaissance médicale est répartie entre plusieurs agents. Leur fonctionnement en parallèle devrait réduire le temps de réponse du système.

3.2. Les agents de RespiDiag

Le modèle comporte quatre agents cognitifs "patho" correspondant chacun à l'une des pathologies considérées dans l'étude : Tuberculose, Asthme, BPCO, et Infections. Rappelons que les agents cognitifs disposent d'un savoir faire et d'une représentation interne du monde sur laquelle ils peuvent raisonner. Ils se basent sur cette représentation pour tenter de réaliser leurs objectifs et communiquent par le biais de messages. Chaque agent "patho" dispose d'un ensemble de connaissances, de deux modes de raisonnement lui permettant un comportement autonome et des capacités de communication. Cela a été modélisé par :

- un module de raisonnement à base de règles de production,
- un module de raisonnement à base de cas et
- un module de communication.

Un cinquième agent cognitif aura la charge de contrôler et de superviser le système; c'est l'agent "superv" qui dispose de :

- un module de contrôle et de supervision,
- un module de gestion de base de données et
- un module de communication.

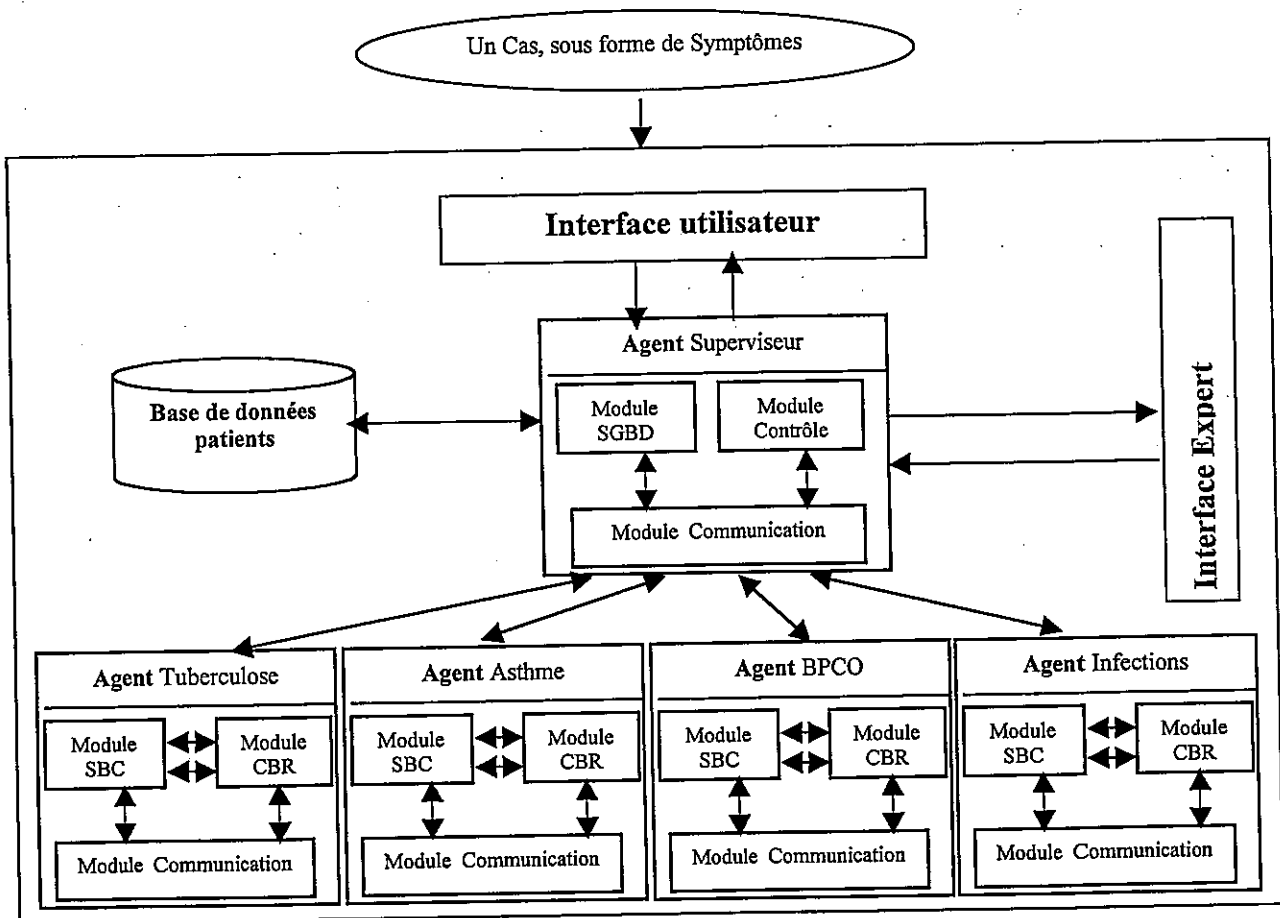


Figure 1 : Architecture du modèle proposé

Le module de gestion de base de données servira pour le suivi des malades dans le temps, ainsi que pour le stockage d'un nouveau cas en manque de données complémentaires.

L'architecture de notre système est présentée dans la figure 1.

3.3. Les interactions entre les agents de RespiDiag

Les agents du système travaillent dans l'objectif de trouver le diagnostic le plus probable en fonction d'une certaine combinaison de symptômes et en se basant sur des connaissances disponibles sous forme de règles de production et de cas résolus et stockés. Comme un malade peut présenter plusieurs diagnostics à la fois, cela fait que les objectifs des agents restent compatibles et ne s'opposent pas. Les informations nécessaires à la décision sont disponibles pour tous les agents et il n'y a pas de conflit pour y accéder. En plus, et vu que la connaissance médicale est répartie entre les agents, ces derniers se trouvent dans un manque de compétence de décision (chacun dispose d'une connaissance partielle). Tout cela traduit que les agents de notre système interagissent en collaboration pour atteindre leurs objectifs.

3.4. Les modules CBR des agents "patho"

3.4.1. La description des cas

Pour chacun de ces agents, le module CBR comporte une base de cas, des métriques de similarité et fera appel à des règles de production du module SBC pour sa phase d'adaptation. L'organisation de la mémoire est linéaire, et le cas sera pour nous, un vecteur d'attributs - valeurs qui servira à :

- La description de l'examen clinique associé à une consultation pour un patient donné (les symptômes respiratoires qui posent le problème),
- La description de la prise en charge adéquate, à savoir le diagnostic avec son traitement (la solution à ce problème).

En collaboration avec des médecins de la spécialité, une liste de descripteurs représentant les symptômes des maladies a été dégagée. Ces variables sont de nature hétérogène; certaines sont numériques et d'autres sont littérales. A titre d'exemple, nous avons :

- la température : de nature numérique,
 - la toux (sèche ou productive) de nature littérale.
- Ci-dessous, nous citons les descripteurs des quatre pathologies considérées dans l'étude : Tuberculose, Asthme, BPCO et Infections.

Tuberculose : Age, Sexe, Température, Toux, Expectorations, Crachats hémoptoïques,

Amaigrissement, Asthénie, Anorexie, Sueurs nocturnes, Contage tuberculeux, Opacité et BK

Asthme : Age de survenue des symptômes, Dyspnée, Toux sèche, Oppression Thoracique, Absence de tabagisme, EC1, EC2, DEP1, DEP2, EFR1, EFR2 (voir tableau n°1)

Infections : Début brusque des signes, Frissons, Fièvre, douleur thoracique, Toux productive purulente, râles crépitants en foyers, opacité, hyper leucocytose,

B P C O : Sexe, Age, Dyspnée, Oppression Thoracique, Tabagisme, EC1, EC2, DEP1, DEP2, EFR1, EFR2

Nous mentionnons les significations de certains paramètres descripteurs par la suite.

3.4.2. La pondération des descripteurs

Sachant que l'importance de ces paramètres descripteurs varie d'un diagnostic à un autre, et que même pour un diagnostic donné, des variables sont plus déterminantes que d'autres, une estimation de cette importance a été établie par les médecins et traduite dans le système sous forme de coefficients. Ces derniers ont permis le calcul des taux de participation du paramètre dans le diagnostic qui vont servir de poids dans le calcul des similarités entre les cas.

Le descripteur BK seul est un paramètre de détermination. La présence d'une valeur positive pour la BK confirme à 100% le diagnostic de Tuberculose. En cas d'absence de cette information, l'ensemble des autres descripteurs se partagent la valeur totale du taux.

3.4.3. Le calcul de similarité entre les cas

Bisson définit dans [Bisson.00] l'identification comme étant un processus ayant pour but de trouver à quel objet, un objet inconnu ressemble le plus. L'une des méthodes qui maximisent la similarité avec cet objet inconnu est bien la méthode des K-plus proches voisins. Elle a prouvé son efficacité notamment dans le raisonnement à base de cas où le paramètre K est égal à 1.

La similarité de deux cas dépend des similarités de leurs attributs pondérés. La formule suivante a été adoptée pour le calcul de similarité entre deux cas X et Y.

$$\text{Sim}_{Ag}(X, Y) = \frac{\sum_{i=1}^n C_i \times \text{Sim}(p_i(x), p_i(y))}{\sum_{i=1}^n C_i}$$

Où C_i représente le coefficient (poids) du paramètre p_i dans l'agent Ag , et n représente le nombre total des paramètres dans le même agent.

Pour tout paramètre de nature symbolique (où la valeur est égale à oui/non, ou bien masculin/féminin), la similarité entre deux cas, est égale à 1 s'ils présentent la même valeur, et elle est égale à 0 dans le cas contraire.

Ci-dessous, nous présentons quelques heuristiques de similarité pour les paramètres de nature numérique :

Pour le paramètre **âge**, nous avons supposé similaires deux cas s'ils présentent un écart de 3 ans au plus, ainsi :

$\text{Sim}(\text{âge}(x), \text{âge}(y)) = 0$ si $|\text{âge}(x) - \text{âge}(y)| > 3$
 $\text{Sim}(\text{âge}(x), \text{âge}(y)) = 1$ si $|\text{âge}(x) - \text{âge}(y)| \leq 3$ ans

Pour le paramètre **Evolution des signes**, nous nous intéressons à savoir si l'apparition des signes date de plus de 3 semaines. Ainsi :

$\text{Sim}(\text{Evol}(x), \text{Evol}(y)) = 0$ si $(\text{Evol}(x) \geq 3 \text{ et } \text{Evol}(y) < 3)$ ou $(\text{Evol}(x) < 3 \text{ et } \text{Evol}(y) \geq 3)$
 $\text{Sim}(\text{Evol}(x), \text{Evol}(y)) = 1$ si $(\text{Evol}(x) \geq 3 \text{ et } \text{Evol}(y) \geq 3)$ ou $(\text{Evol}(x) < 3 \text{ et } \text{Evol}(y) < 3)$

Pour le paramètre **Température**, nous nous intéressons à savoir si le patient présente une fièvre ou non. Ainsi :
 $\text{Sim}(\text{temp}(x), \text{temp}(y)) = 0$ si $(\text{temp}(x) \leq 38 \text{ et } \text{temp}(y) > 38)$ ou $(\text{temp}(x) > 38 \text{ et } \text{temp}(y) \leq 38)$
 $\text{Sim}(\text{temp}(x), \text{temp}(y)) = 1$ si $(\text{temp}(x) \leq 38 \text{ et } \text{temp}(y) \leq 38)$ ou $(\text{temp}(x) > 38 \text{ et } \text{temp}(y) > 38)$

3.4.4. L'adaptation

Durant la phase d'adaptation, les deux modules CBR et SBC entrent en interaction et travaillent en collaboration pour réaliser cette tâche. Le CBR demande la liste des recommandations (médicament et doses) pour le nouveau cas déjà connu par le SBC. Puis, il procède à une comparaison entre les données reçues et le traitement prescrit pour l'ancien cas.

Dans le cas où les deux traitements (ancien et recommandé) sont semblables, l'adaptation se réduit à un transfert de l'ancienne solution au nouveau cas. Sinon, le CBR envoie l'ancien traitement à l'SBC pour voir s'il peut être prescrit au nouveau malade. En cas d'affirmation le SBC envoie les doses recommandées, sinon le CBR retient le premier traitement dicté par le SBC et le considère comme solution adaptée pour le nouveau cas.

3.5. Les modules SBC des agents "patho"

Chaque module SBC fonctionne comme un petit Système Expert qui raisonne sur un ensemble de règles

de production menant uniquement vers le diagnostic de l'Agent concerné. Il est constitué d'une base de règles, d'une base de faits et d'un moteur d'inférence qui raisonne selon une stratégie de chaînage mixte.

Des règles de productions ont été établies. Elles permettent le diagnostic des quatre pathologies : Tuberculose, Asthme, BPCO et Infections et proposent également des thérapies pour chaque diagnostic (liste de médicaments avec doses et durées de traitement). Et vu que notre système présente une architecture distribuée, ces règles ont été réparties entre les quatre modules SBC des différents agents. Ainsi, leur moteurs d'inférences fonctionneront en parallèle, ce qui garantit un gain de temps d'exécution et une rapidité dans le temps de réponse du système.

3.6. Fonctionnement du système

A l'arrivée d'un nouveau cas, l'agent "superv" envoie les paramètres vers les quatre agents "patho". Ces paramètres sont reçus en tant que faits par les modules SBC et en tant que cas par les modules CBR. Deux résultats sont attendus de chaque agent ; une réponse binaire (une reconnaissance ou non) de la part de chaque module SBC et un score de similarité de la part de chaque module CBR.

Notons ici, que l'étape d'adaptation n'est pas entamée juste après la sélection du meilleur cas mais reportée après la décision du "superv". Dans l'habitude, un système CBR enchaîne directement sur la phase d'adaptation dès que le meilleur cas est trouvé. Dans notre modèle, nous disposons de quatre bases de cas, et chaque agent est censé nous donner le meilleur score trouvé dans sa base.

L'agent "superv" décidera par la suite "le meilleur des meilleurs", et désignera lequel des agents « patho » devra entamer la tâche d'adaptation. Cette décision est principalement basée sur une comparaison des réponses des modules SBC et des scores de similarités reçus.

Un module SBC peut ne pas reconnaître le cas et répond négativement. Cela ne signifie pas forcément qu'il faut ignorer le diagnostic correspondant, car la réponse négative peut être due à un manque de données pertinentes dans les paramètres entrants. Dans ce cas, le superviseur tranchera en fonction du score de similarité envoyé par le même agent.

Une fois le diagnostic décidé, l'agent correspondant est recontacté pour qu'il enchaîne sur la phase l'adaptation de la solution trouvée.

Cette solution est envoyée vers l'agent superviseur. Ce dernier la présente à l'utilisateur pour validation puis, un stockage du nouveau cas dans la base correspondante est effectué.

Conclusion

Dans ce travail, l'IAD et la médecine se rencontrent pour apporter chacune un intérêt pour l'autre. Nous avons choisi la santé respiratoire comme champs d'expérimentation, et nous avons conçu un système modélisant le raisonnement médical. Il combine deux modes de raisonnement de l'IA, à savoir le raisonnement basé règles et le raisonnement basé cas dans la même entité. Cette combinaison permettra d'exploiter les avantages des deux paradigmes. Dans l'objectif d'améliorer les performances du système, la notion du parallélisme a été introduite en répartissant la connaissance médicale entre plusieurs agents cognitifs. Chaque agent dispose de moyens d'interactions avec son environnement, et ses deux modules CBR et SBC interagissent entre eux pendant la phase d'adaptation du cycle CBR.

Références

- [Nunn.00] Paul Nunn, Scientific Working Group on Tuberculosis, Geneva Feb 9-11, 2000
- [Benali et al.04] Rachid Benali et al., la santé respiratoire. Rapport technique, Service Pneumologie de l'Hôpital Dorban Annaba, 2004.
- [Bisson.00] Gilles Bisson, Induction symbolique/numérique à partir de données, chapitre xx, la similarité : une notion symbolique/numérique, E.Diday, Y. Kodratoff, P.Brito, M Moulet, Toulouse : cepaduès, Edition 2000, pp 169-201.
- [Heinisch et al.99] Roberto Henrique Heinisch, Rosina Weber, Alejandro Martin et Ricardo M.Barcia, Industrial Engineering and Cardiology Service of University Hospital, Florianopolis, SC, Brazil
- [Schmidt et al.90] Schmidt, H.G.; Norman, G.R.; Boshuizen, H.P.A. A cognitive perspective on medical expertise: Theory and implications. Academic Medical, 1990; 65: 611-621.

Un Système Tuteur Utilisant un Compagnon Perturbateur

Tahar Bouhadada - Med Tayeb Laskri

Département d'Informatique - Université Badji-Mokhtar Annaba

B.P. 12 Annaba 23000 Algérie

Email: : Bouhadada.t@voila.fr, laskri@univ-annaba.org

Résumé :

Ce papier présente l'architecture d'un système tuteur intelligent (STI) basé sur la stratégie d'apprentissage du double test (DTL) utilisant deux compagnons comme co-apprenants où un des deux est un perturbateur. Le système appelé DB-TUTOR est proposé pour l'enseignement des bases de données relationnelles pour des étudiants de 3^{ème} année du cycle ingénieur.

Abstract :

This paper outlines the architecture of an intelligent tutoring system (ITS) based on the double test learning (DTL) strategy and using two companions as co-students where one of both is a troublemaker. The system called DB-TUTOR is proposed to teaching the relational databases for the students of the 3rd year of the graduate cycle.

MOTS-CLEFS : Système Tuteur Intelligent, Modèle de l'apprenant, Compagnon, Perturbateur, Stratégie DTL.

1. Introduction

Plusieurs Systèmes Tuteurs Intelligents (ITS) utilisant un compagnon-d'apprentissage-machine appelé Learning Companion System (LCS) ont été développés ces dernières années. Ces systèmes impliquent trois agents: l'Enseignant, l'Apprenant et un Compagnon. Le principe de ces systèmes utilise l'idée que l'interaction sociale a une influence sur le développement cognitif. L'idée de base est que l'Apprenant peut apprendre de ses erreurs et peut tirer profit des erreurs du compagnon. Le compagnon peut être perçu comme un assistant, et d'autre fois, comme un perturbateur. L'objectif du compagnon d'apprentissage est de stimuler la compétition et la collaboration à travers la démonstration.

Dans ce papier, nous esquissons une classe d'environnement de l'apprentissage, appelée Learning Companion System (LCS) qui est un Système Tuteur Intelligent et où nous introduisons deux compagnons comme co-apprenants dans la session d'apprentissage où un des deux est un perturbateur.

Nous décrivons l'architecture du système, basée sur la stratégie du double test (DTL) et utilisant deux compagnons comme co-apprenants. Le système appelé DB-TUTOR est proposé pour l'enseignement des Bases de données pour les étudiants de la 3^{ème} année du cycle ingénieur.

Nous décrivons la structure du modèle-apprenant qui sera implémenté et nous donnerons les perspectives de développement de notre système futur.

2. Stratégie d'apprentissage à double test

2.1. Idée de base

Les expériences [2] ont montré que deux enfants travaillant ensemble peuvent exécuter et réaliser des tâches qui ne peuvent être exécutées par des enfants travaillant seuls. Une expérience subséquente [3] indique qu'il y a plus de progrès quand des enfants, avec des stratégies cognitives différentes, travaillent ensemble que quand ils travaillent séparément, avec les mêmes stratégies. Aussi, non seulement le moins avancé progresse mais l'enfant le plus avancé évolue aussi à partir des réactions de son compagnon.

En effet, l'influence de l'interaction sociale sur le développement cognitif individuel a été étudiée et observée par plusieurs chercheurs dans des régions différentes.

Dans son travail, Vygotsky affirme que les interactions sociales jouent un rôle fondamental

dans la construction des structures cognitives internes [4].

Une activité d'enseignement en science implique d'un point de vue didactique, d'une manière générale, de prendre en considération l'évolution de la conception de l'apprenant (dimension cognitive), la motivation (dimension émotive) ainsi que des relations avec ses collègues et /ou avec l'enseignant (dimension sociale) [5].

Pettito [6] plus tard, montre que l'approche prise par une paire d'apprenants devant une tâche donnée peut être souvent qualitativement différente de l'approche prise par l'un ou l'autre apprenant seul.

La structure d'un LCS implique trois agents (Figure 1), à savoir, l'apprenant humain, le compagnon-apprenant-machine et l'ordinateur-enseignant [1]. Le rôle de l'ordinateur-enseignant est de fournir des exemples, des indications, et des commentaires à l'apprenant et au compagnon. L'objectif du compagnon d'apprentissage est de stimuler la collaboration à travers la compétition.

2.2. Rôle des agents

2.2.1. Rôle du tuteur

La tâche du tuteur est de produire des leçons et des exercices, expliquer les exemples, définir la stratégie d'apprentissage, définir la stratégie pédagogique, dialoguer avec l'apprenant et lui fournir des arguments et des justifications pour les solutions données.

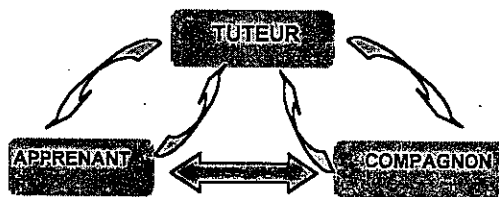


Figure 1 : Architecture d'un LCS

2.2.2. Rôle de l'apprenant humain

L'objectif de l'apprenant humain est d'apprendre et acquérir la connaissance. Pour atteindre cet objectif, l'apprenant doit participer aux différentes phases de la stratégie d'apprentissage. En premier lieu, il doit suivre la phase apprentissage. En second lieu, il doit prendre le rôle du compagnon dans la deuxième phase. Et en fin, il doit être évalué comme son compagnon.

2.2.3. Rôle du compagnon

Dans un système LCS, le rôle du compagnon peut être parfois un concurrent, et d'autre fois un perturbateur. Dans certains autres systèmes, l'apprenant peut trouver aide et assistance chez le compagnon[7][8].

Pendant la session d'apprentissage, le compagnon peut être amené à jouer plusieurs rôles. Il coopère avec l'apprenant en lui permettant de bénéficier de ses réponses (comportement). Un autre rôle est de s'assurer que la connaissance de l'apprenant est renforcée pendant la session d'apprentissage. L'apprenant doit justifier sa réponse correcte quand celles de ses compagnons sont erronées. Le compagnon est aussi utilisé pour la confiance en soi de l'apprenant quand il répond incorrectement aux questions ambiguës.

2.2.4. Un système LCS avec deux compagnons

Dans ce système, nous introduisons un deuxième compagnon comme un perturbateur [15]. La tâche de ce compagnon est d'introduire des perturbations dans la session durant la phase Pre-test1 pour évaluer l'assurance et la confiance de l'apprenant. L'idée est de tester la conviction de l'apprenant quand il est évalué par le système. Une grande partie des réponses du perturbateur est incorrecte volontairement.

2.2. Description

Une session d'apprentissage typique qui utilise la stratégie DTL débute avec une phase Pré-Test dans laquelle un modèle-apprenant initial est créé. Comme pour la deuxième phase, les co-apprenants bénéficient du même apprentissage que l'apprenant-humain, ainsi les trois ont le même niveau de connaissance.

Dans la troisième phase (Post-Test1), le tuteur évalue les co-apprenants. L'apprenant humain se trouvera dans la place d'un observateur actif où il suivra la session de questions/réponses entre le tuteur et les co-apprenants. L'apprenant a aussi accès à un bloc-notes où il peut mentionner toutes observations. A chaque fois que les co-apprenants donnent la

solution d'un problème, le tuteur évalue leurs réponses. Si leurs réponses sont incorrectes et que celle de l'apprenant humain est correcte, ce dernier doit justifier et expliquer sa réponse aux co-apprenants. Quand les co-apprenants auront terminé la phase Post-Test1, le tuteur alors se tourne vers l'apprenant humain et la phase Post-Test2 commence. Ici le bloc-notes de l'apprenant est retiré, et par conséquent, il n'a accès qu'à sa mémoire seulement et la connaissance qu'il a acquise récemment à travers les réponses des co-apprenants. Et en fin, le tuteur évalue ses réponses.

2.3. Les phases

2.3.1. La 1^{ère} phase: La phase Pré-Test

A l'entrée de l'apprenant dans le système, Il est informé sur toutes les différentes phases qu'il va suivre (Pré-Test, Apprentissage, Post-Test1 et Post-Test2) et il lui est aussi donné une brève présentation au sujet de ses co-apprenants. Une fois la présentation faite, la phase Pré-Test commence. Le tuteur donne une série de questions auxquelles l'apprenant doit répondre. Le Tuteur évalue alors ses réponses sans qu'aucune explication sur les réponses incorrectes ne lui soit fournie. Le système et le Tuteur ont connaissance du score de l'apprenant obtenu dans cette phase. Une fois qu'il ait répondu à toutes les questions, le tuteur informe alors l'apprenant du début de la phase d'apprentissage.

2.3.2. La 2^{ème} phase : La phase apprentissage

Durant l'apprentissage, le tuteur présentera une série d'exercices suivis immédiatement de leurs solutions et des explications. Pendant l'enseignement, l'apprenant et les co-apprenants suivent simplement ce que le tuteur leur apprend.

2.3.3. La 3^{ème} phase: La phase Post-Test1

Quand la session d'enseignement est terminée, l'apprenant peut prendre des notes pendant que le tuteur teste les co-apprenants. En fait pendant ce temps, l'apprenant a en sa possession un bloc-notes sur lequel il peut mentionner n'importe quelle remarque qui puisse l'aider à résoudre les exercices et répondre correctement aux questions.

L'apprenant humain observe les questions et les réponses de ses pairs. Il peut même connaître leurs scores. L'apprenant va sans aucun doute prendre en considération les réponses des co-apprenants avant de donner sa propre réponse. Une fois toutes les réponses soumises, le tuteur donne alors la bonne réponse à l'apprenant. Seules les réponses des co-apprenants sont évaluées. Dans la plupart des cas, les réponses des co-apprenants ne sont pas identiques. Si leurs réponses sont correctes, leurs scores seront augmentés. Dans le cas où les co-apprenants se sont trompés et que la réponse de l'apprenant est correcte, les co-apprenants demanderont à l'apprenant de leur fournir des explications, ce qui renforcera sa propre connaissance. Nous devons mentionner que le tuteur a accès aux réponses de l'apprenant. Le score de ce dernier est augmenté chaque fois que ses réponses sont correctes, cependant, il perd des points si l'argumentation n'est pas adéquate. A chaque fois que l'apprenant répond incorrectement, le tuteur fournira aux apprenant des explications sur la solution. Les détails des explications du tuteur dépendent du niveau de connaissance de l'apprenant.

2.3.4. La 4^{ème} phase: La phase Post-Test2

A la fin de la phase Post-Test1, les co-apprenants disparaissent et le tuteur commence alors la phase Post-Test2 où l'apprenant humain sera testé. Le bloc-notes n'est plus accessible pour consultation et, en aucun cas le tuteur ne doit fournir de réponses ou d'explications. Plusieurs des questions présentées dans cette phase sont reprises de la phase Post-Test1. Parmi elles, des questions ambiguës engendrées par le système. Le reste de la matière comprend des questions qui ont été déjà présentées dans la phase Post-Test1 avec quelques variantes et d'autres totalement nouvelles. A la fin de cette session, le tuteur donne les réponses et les explications seulement aux questions auxquelles l'apprenant a mal répondu.

3. DB-TUTOR

Le système DB-TUTOR comprend les modules traditionnels d'un STI (Figure 2):

- Un module expert;

- Un module tuteur;
 - Un modèle-apprenant;
 - Un module pédagogique
 - Une interface utilisateurs;
- et un composant d'un LCS: le module co-apprenants.

3.1. Architecture de DB-TUTOR

Le système DB-TUTOR est un Système Tuteur Intelligent conçu pour permettre d'apprendre les bases de données aux étudiants de la 3^{ème} année du cycle ingénieur. Le système est organisé essentiellement autour des modules suivants [10](Figure 3):

- La base de la connaissance des concepts et des métaconcepts;
- Un tuteur pédagogique expert qui utilise un générateur de leçons, permet de choisir la stratégie pédagogique selon le profil de l'apprenant;
- Une interface enseignant qui permet à l'enseignant d'insérer, de modifier, de supprimer, ou consulter des concepts ou des métaconcepts;
- Une interface apprenant qui permet de gérer la communication apprenant/machine;
- Une base de profils-apprenants qui contient l'historique des co-apprenants.
- Le module co-apprenant qui génère un comportement simulé des compagnons;

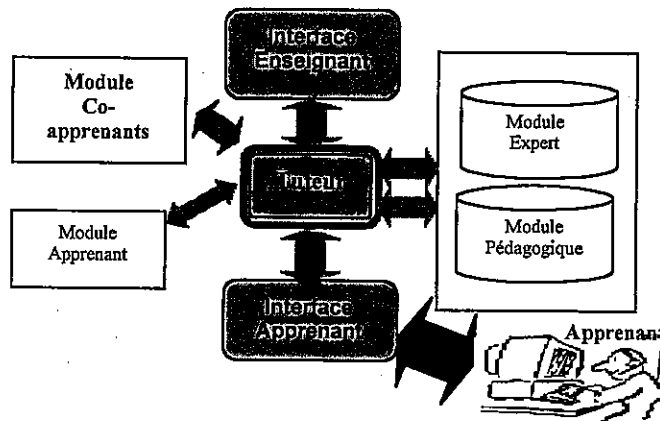


Figure 2. Structure du système DB-TUTOR

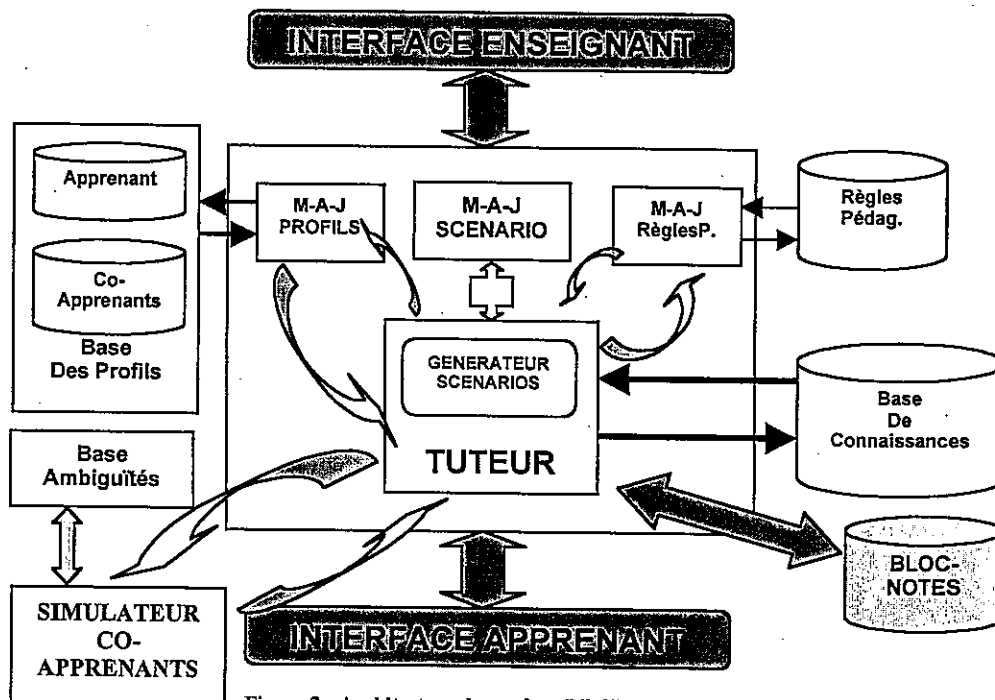


Figure 3 : Architecture du système DB-TUTOR

3.2. Environnement de développement du système DB-TUTOR

La stratégie utilisée par le système DB-TUTOR est basée la méthode d'enseignement par découverte qui utilise des outils multimédias comme moyen pour faciliter l'apprentissage des

bases de données, capable de résoudre les exercices proposés, et permet à l'apprenant de concevoir et modéliser un environnement [10]. Le système est destiné aux étudiants de 3^{ème} année du cycle ingénieur. Le système détermine le concept et le métaconcept à enseigner et les exercices à proposer selon l'objectif et le profil de l'apprenant.

3.3. Objectifs du système

Le système est un système d'aide qui permet d'assister l'enseignant et permet à l'apprenant d'acquérir la connaissance sur les bases de données relationnelles, d'avoir une compétence à résoudre les exercices et la maîtrise du processus de conception. L'objectif de notre projet est la réalisation d'outils permettant d'assister les apprenants dans la conception et la modélisation des bases de données.

3.4. Le domaine à enseigner

L'apprentissage des bases de données est un processus long et assez complexe. Le monde des bases de données se construit petit à petit; chaque notion servant de support aux suivantes.

Cet apprentissage comporte à chaque niveau:

- L'acquisition des notions;
- L'entraînement à l'utilisation de ces notions.

Afin de décrire le domaine des bases de données, nous avons été amenés à diviser le domaine en 3 sous-domaines:

- *Le domaine des prérequis indispensables*
Ces notions sont nécessaires à notre enseignement. Notre système a été conçu pour en permettre l'apprentissage. Ces prérequis sont définis par rapport au niveau de l'apprenant.
- *Le domaine des notions nécessaires*
L'apprenant est déjà familiarisé avec ces notions dont l'enseignement n'est pas le but essentiel de la session. Leur connaissance est pourtant indispensable à la résolution des exercices proposés et leur méconnaissance un facteur de la mauvaise compréhension du domaine.
- *Le domaine enseigné*
L'aspect où l'apport est de notre système sera est le plus important. C'est la connaissance à dispenser à l'apprenant. Il contient tous les concepts et métaconcepts concernant le domaine des bases de données relationnelles.

3.4.1. La connaissance

La connaissance nécessaire pour la résolution des exercices est variée et définie par des concepts. Les concepts sont regroupés dans des métaconcepts. La connaissance du système DB-TUTOR est composée actuellement de 294 concepts regroupés dans 132 métaconcepts (Tableau 1 et Tableau 2).

#	Concepts
.....
102	La dépendance fonctionnelle
103	La dépendance fonctionnelle totale
104	La dépendance fonctionnelle directe
105
.....

Tableau 1. Liste des concepts

#	Métaconcepts	Concepts	Prérequis
.....
52	Le processus de décomposition	102,103	47,48
53	La normalisation	102,103	52
54	La première forme normale	102	49
55	La deuxième forme normale	102,103	52,54
.....

Tableau 2. Liste des métaconcepts

3.4.2. Le bloc-notes

Le bloc-notes contient toutes les observations mentionnées par l'apprenant durant son observation de l'apprentissage des co-apprenants. L'apprenant utilise le bloc-notes pour noter toutes remarques durant l'évaluation des co-apprenants dans la phase Pré-Test1. Ce bloc-notes contient en particulier les erreurs commises par les co-apprenants et toute démonstration erronée fournie dans la résolution des exercices. L'apprenant doit alors mentionner l'index de l'exercice

où la réponse donnée est erronée ainsi que la démonstration l'accompagnant donnée par les co-apprenants. Les notes contenues dans le bloc-notes sont utilisées aussi dans l'évaluation de l'apprenant.

3.4.3. Le modèle-apprenant

Les modèles de l'apprenant sont un composant essentiel des systèmes tuteurs intelligents. Les recherches faites sur les modèles se sont centrées sur deux structures : le modèle overlay et le modèle des erreurs. Le développement d'un modèle pour représenter les aptitudes d'un apprenant est complexe et dépend du domaine d'apprentissage. Le module modèle-apprenant est chargé d'analyser les réponses de l'apprenant et de gérer la structure du modèle afin de pouvoir suivre l'évolution et la progression de l'apprenant.

Le modèle de l'apprenant ou le profil de l'apprenant doit traduire à chaque instant le niveau de connaissance de l'apprenant et des co-apprenants. Il comprend quelques règles psychopédagogiques, et des stratégies pédagogiques optimales. Il est élaboré au fur et à mesure de la session par observation et analyse des interactions du tuteur/apprenant [11].

Le modèle de l'apprenant représente l'état de sa connaissance. La plupart des systèmes tuteurs intelligents (STI) considèrent l'état de la connaissance de l'apprenant comme un sous-ensemble de la connaissance de l'expert [12].

La connaissance de modélisation de l'apprenant et l'apprentissage de son comportement impliquent deux processus: l'enregistrement des connaissances acquises par l'apprenant, en particulier, ses lacunes, et identifier la stratégie d'apprentissage à partir du protocole de l'apprenant. C'est pourquoi, le modèle est construit en accumulant des métriques sur les réalisations et le comportement de l'apprenant. Ce modèle conserve aussi une trace de l'erreur conceptuelle principale détectée par l'étudiant, des observations mentionnées durant la phase Post-Test1 et de l'intervention du professeur.

La modélisation de l'apprenant doit prendre en considération le comportement des co-apprenants pendant la phase Post-Test1 pendant que le tuteur teste les co-apprenants.

4. Le modèle-apprenant de DB-TUTOR

Le tuteur génère le modèle de l'apprenant. Les niveaux traduisent le modèle de l'apprenant de DB-TUTOR. 9 niveaux numérotés de 0 à 8 ont été définis pour les apprenants et les compagnons. Des jetons sont délivrés à chaque fin d'exécution d'un scénario en fonction du comportement de l'apprenant[10].

4.1. Structure du modèle

Le modèle est défini par des niveaux. Un niveau est décrit par [10]:

▪ *Un prérequis*

C'est l'état des connaissances (déjà) acquises par l'apprenant. Il correspond au profil d'un apprenant du niveau inférieur.

Le prérequis est défini par 2 valeurs : Moyen (=) ou Bon (+)

Un apprenant ne peut postuler pour le niveau immédiatement supérieur que si le ou les prérequis ont été acquis avec un score Moyen ou Fort.

▪ *Les métaconcepts*

Ce sont les connaissances (les notions) à dispenser et qui caractérisent le niveau courant de l'apprenant. Les notions que l'apprenant cherche à acquérir. Ce sont les leçons.

▪ *L'évaluation*

L'évaluation des réponses est assurée par des règles de production déclenchées par chaînage avant [14].

SI Réponse ALORS Objectif_Pédagogique (Crédit, Débit)

Les valeurs Débit et Crédit représentent les jetons à ajouter ou à retrancher du score courant

selon l'objectif.

L'évaluation est faite sur la base des connaissances acquises du niveau courant et sur la base des observations mentionnées sur le bloc-notes de l'apprenant. Cette évaluation est caractérisée par une série d'exercices qui permet de donner un profil de l'apprenant à la fin du scénario. Cette évaluation permet à l'apprenant d'accéder au niveau supérieur.

Un ensemble d'exercices est associé à chaque niveau. Chaque exercice est défini par un degré de difficulté et un certain nombre de jetons est délivré pour chaque succès.

- Facile (-) nombre de jetons = 1
- Moyen (=) nombre de jetons = 2
- Difficile (+) nombre de jetons = 3

Pour un niveau donné, le profil de l'apprenant est désigné par :

Faible: si le nombre de jetons acquis est inférieur au nombre de jetons requis.

Moyen: si le nombre de jetons acquis est égal au nombre de jetons requis;

Fort: si le nombre de jetons acquis est supérieur au nombre de jetons requis.

Le profil de l'apprenant est déterminé par la fonction suivante:

PROFILE (CodStud, Level, Score)

Où:

- CodStud est l'identification de l'apprenant;
- Level est le niveau de l'apprenant;
- Score est le score obtenu pour chaque niveau;

▪ *Définition des niveaux*

Neuf niveaux ont été définis et chaque niveau est décrit par:

- Le nombre de jetons requis;
- Les concepts prérequis;
- Les concepts à dispenser;
- Les exercices
- Les observations du bloc-notes

LEVEL(CodStud, ListCptPreq, ListCpt, ListEx, NotBok, NbrTokReq)

Où:

- ListCptPreq est la liste des concepts prérequis;
- ListCpt est la liste des concepts;
- ListEx est la liste des exercices;
- NbrTokReq est le nombre de jetons requis;
- NotBok les observations;

5. Le scénario

Un scénario constitue une composante d'une session d'enseignement. Il inclut un ensemble de séquences d'enseignement et un ensemble d'exercices.

Le système inclut 3 catégories des séquences:

- Des séquences d'enseignement (leçons);
- Des séquences d'aide ;
- Des stratégies pédagogiques.

6. Le score

Le calcul du score final est basé sur le nombre de jetons acquis dans les différentes phases. Ce score prend aussi en considération les observations sur le bloc-notes de l'apprenant et son comportement durant les différentes étapes:

SCORELEVELi (CodStud, Score1, Score2, Score3)

Où:

- SCORELEVELi est le score obtenu pour le niveau i ;

- Score1 est le score obtenu dans la phase Pré-Test;
- Score2 est le score obtenu dans la phase Post-Test1 ;
- Score3 est le score obtenu dans la phase Post-Test2;

Cette fonction donne pour le niveau courant le comportement de l'apprenant durant les différentes phases d'évaluation et la valeur est obtenue par:

$$\text{SCORELEVEL} = \text{Score1} + \text{Score2} + \text{Score3}$$

7. Conclusion

Nous avons présenté dans ce papier un environnement interactif d'apprentissage par ordinateur utilisant un compagnon perturbateur. Nous avons décrit la stratégie d'enseignement DTL introduisant deux compagnons dans une session de l'apprentissage. Nous avons défini le rôle et la fonction de chaque agent implémenté. Nous avons décrit l'architecture du système à réaliser dans un environnement multimédia.

Le système est réalisé avec le langage JACK Intelligent Agent sur une plate-forme JAVA. Une validation du système sera réalisée prochainement avec une population choisie parmi les étudiants de 3^{ème} année du cycle ingénieur.

Références Bibliographiques

- [1] CHAN.T. W.; BASKIN. A. B. : "Studying with Prince' The computer as a learning companion". Intelligent Tutoring Systems, ITS'88, Montréal, Canada, Juin 1-3, 1988.
- [2] DOISE. W.; MUGNY. G. : "Social interaction and the development of cognitive operations". European journal of social psychology, 5(3), p.367-383, 1975.
- [3] MUGNY. G.; DOISE. W. : "Socio-cognitive conflict and structure of individual and collective performances". European journal of social psychology, 8, p. 181-192, 1978.
- [4] VYGOTSKY. L. S. : "Mind in Society", Cambridge, MA, Harvard university Press, 1978.
- [5] CHOPLIN. H. ; GALISSON A.; LEMARCHAND S. : "Hypermédias et pédagogie: comment promouvoir l'activité de l'élève ?" Hypermédias et Apprentissage, 4^{ème} colloque, Poitiers, France, 15-17, October, 1998.
- [6] PETTITO. A. : "Collaboration in problem solving". University of Rochester, 1985.
- [7] AIMEUR. E. : "Application and Assessment of Cognitive-Dissonance Theory in the Learning Process". Journal of Universal Computer Science, 4 (3), 1996.
- [8] AIMEUR. E.; FRASSON. C. : "Analyzing a new learning strategy according to different knowledge levels", Computers and Education, an international, 27 (2), p.115-127, 1996.
- [9] FAHMI. M.; AIMEUR. E. : "RACSY: an intelligent tutoring system based on the double test learning strategy". 5th Maghrebian Conference on Software Engineering and Artificial intelligence, MCSEAI'98, December, 7-10, Tunis, Tunisia, 1998.
- [10] BOUHADADA. T.; LASKRI. M.T. : "SACEPAF+: A learning environment of elementary calculus intelligently aided by a computer". Current Trends in Computer Science and Information System, CTCISIS'96, Jordan, July, 9-10, 1996.
- [11] QUAFARFOU. M.; PREVOT. P. : "The design of an ITS dedicated to industrial proceedings". IEEE, International Conference on Developing and Managing, p.154-158, 1993.
- [12] BENSEBAA. T.; SERIDI. H. : "Architecture d'un hypermédia éducatif". Le journal du Multimédia, N°20, 1997.
- [13] AYALA. G.; YANO. Y. : "Intelligent agents to support the effective collaboration in a CSCL environment". World conference on educational communications, June 17-22, 1996, Boston, Massachusetts, USA.
- [14] BENSEBAA. T.; LAFIFI. Y. : "Architecture d'un hypermédia éducatif et coopératif". Actes du 5^{ème} Colloque Africain sur la recherche en informatique. CARI 2000, Antananarivo, Madagascar, 16-19, Octobre, 2000.
- [15] FRASSON. C; AIMEUR. E.: "A comparison of three learning strategies in intelligent tutoring systems". Journal of Educational Computing Research, Vol 14, 1996.

VOUZZAVEDIBISAR

Fantaisie

Edmond Bianco

Quand je tirais le dernier bord, mon petit voilier s'approcha lentement de la côte, je donnais un petit coup de barre pour la longer en cherchant où aborder. Et en même temps je lâchais le foc et la grand'voile. La terre à une dizaine de mètres par trois mètres de fond, une étroite plage de sable, mais aussi pas mal de petits rochers, je ne pouvais aborder tout de suite. Je retendis légèrement le foc, et je pris un peu de vitesse. La côte me semblait bien monotone, il m'aurait fallu un petit havre, un recoin, une petite jetée, une quelconque protection quoi ! J'étais fatigué et j'aurais bien voulu passer la nuit au calme, le temps ne menaçait pas, mais on ne sait jamais avec la mer, le moindre petit coup de vent et le temps de se réveiller on est dressé à la côte.

