

Apporter de la flexibilité dans l'observation d'une activité pédagogique

Gligor-Calin Loghin, Thibault Carron, Jean-Charles Marty

▶ **To cite this version:**

Gligor-Calin Loghin, Thibault Carron, Jean-Charles Marty. Apporter de la flexibilité dans l'observation d'une activité pédagogique. Jun 2007. hal-00162138

HAL Id: hal-00162138
https://hal.science/hal-00162138

Submitted on 12 Jul 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Apporter de la flexibilité dans l'observation d'une activité pédagogique

Gligor-Calin Loghin* **, Thibault Carron*, Jean-Charles Marty*

** SYSCOM - Université de Savoie*
Campus Scientifique
73376 Le Bourget du Lac - France
*** Université Technique de Cluj-Napoca*
400027 Cluj-Napoca - Roumanie

Gligor-Calin.Loghin@univ-savoie.fr
Thibault.Carron@univ-savoie.fr
Jean-Charles.Marty@univ-savoie.fr

RÉSUMÉ. *Ce travail concerne l'observation d'activités pédagogiques dans un environnement numérique de travail. Le comportement de l'utilisateur est observé au moyen de traces numériques brutes obtenues durant son activité. Dans cet article nous présentons une approche d'observation flexible à base d'agents logiciels distribués. Afin de permettre le changement de l'objectif d'observation en cours de déroulement de l'activité pédagogique, nous avons défini plusieurs types d'agents observateurs ainsi qu'un langage de configuration des agents adapté à l'observation. Cette approche a été mise en œuvre dans le cadre d'une expérimentation dont nous nous servons pour illustrer notre article.*

MOTS-CLÉS : *observation, traces, transformation de traces, agents d'observation, flexibilité de l'observation, EIAH*

1. Introduction

Dans une classe, un enseignant est continuellement à la recherche de signes qui lui permettent d'appréhender l'état de compréhension des apprenants. Un retour sur ce qui se passe pendant la séance lui permet d'adapter son scénario pédagogique en rajoutant par exemple une séquence explicative supplémentaire ou un nouvel exercice destiné à clarifier la notion présentée.

Durant la séance d'apprentissage, les objectifs d'observation de l'enseignant peuvent varier d'une activité à une autre ou même pendant l'activité. Par exemple, observer les interactions des apprenants peut être très enrichissant durant une activité collaborative et moins signifiant le reste du temps. Enfin, pour chaque situation observée, l'enseignant peut vouloir réagir différemment selon l'apprenant concerné.

Nous nous intéressons à l'observation dans un EIAH. Dans notre démarche, nous souhaitons fournir à l'enseignant un ensemble d'outils qui lui permettent d'observer mais aussi de réagir en fonction de situations particulières observées : un étudiant est en retard, les étudiants en difficulté n'ont pas consulté un document recommandé, etc. Toutes ces observations lui permettent d'adapter ses actions en temps réel et par conséquent, la session d'apprentissage. Par exemple, l'enseignant peut décider d'ouvrir une session de communication avec un apprenant en difficulté ou d'envoyer une clarification à tous les apprenants en constatant des difficultés générales.

Dans un EIAH, l'observation du comportement peut s'appuyer sur l'analyse des « traces » laissées par l'activité des apprenants. Les traces collectées permettent d'observer la mise en place du scénario pédagogique proposé dans la séance [IKSAL & CHOQUET 05] mais aussi, par l'identification d' « épisodes » annotés, la réutilisation de l'expérience d'apprentissage [CHAMPIN et al. 04]. Enfin, les traces sont généralement nombreuses, provenant des sources différentes et ayant différents niveaux de granularité. Par conséquent, le processus d'observation à base de traces suppose généralement, hors la collecte de traces, la transformation (fusion, sélection, structuration) de traces et la visualisation de traces [SETTOUTI et al. 06].

Ces différents aspects sont étudiés au sein de notre équipe. Nous avons étudié la possibilité d'exploiter différentes sources de traces, en mettant en évidence la richesse d'une telle approche [MARTY et al. 07]. Des solutions ont été proposées pour la réduction du volume des traces collectées et pour le changement du niveau de granularité des traces résultantes [COURTIN & TALBOT 06]. Enfin nous avons proposé des visualisations afin de faciliter l'analyse de l'activité des apprenants a posteriori ou au cours de l'activité [FRANCE et al. 06].

L'approche d'observation présentée ici consiste à gérer les différentes étapes d'observation, durant le déroulement de la séance d'apprentissage, par le moyen d'un ensemble d'agents logiciels distribués [CARRON et al. 06]. Chaque agent a des compétences spécifiques et peut coopérer avec les autres agents pour réaliser une tâche d'observation complexe, approche classique dans les Systèmes Multi-Agents

[CASTERAN et al. 00]. Le point qui nous intéresse ici concerne l'utilisation d'un langage agent facilitant la description de la coopération et de la (re)configuration des agents en vue d'observation dans un cadre pédagogique.

Nous proposons dans cet article, un moyen flexible de mettre en place une observation adaptée aux différentes situations d'une activité pédagogique, basée sur un langage de configuration des agents « observateurs ».

Afin de bien comprendre notre objectif, nous décrivons dans la section suivante, une expérimentation en situation réelle que nous avons effectuée en nous appuyant sur notre système d'observation. Nous présentons, ensuite, l'architecture agent pour l'observation, suivie de la justification de la nécessité d'un langage de configuration agent adapté à l'observation. Pour terminer, nous décrivons le langage lui-même en l'illustrant par quelques exemples concrets issus de l'expérimentation.

2. Une expérimentation pédagogique

2.1. Description de l'expérimentation

L'expérimentation s'est déroulée avec un public de 36 étudiants en 3ème année à l'Université Technique de Cluj-Napoca (Roumanie). Pédagogiquement, il s'agissait d'une séance de travaux pratiques d'une heure trente sur la programmation web. Cette expérimentation visait une meilleure perception de l'activité en cours par l'enseignant tuteur, et ceci en temps réel.

Durant la séance, les étudiants avaient pour consigne de réaliser un scénario pédagogique proposé, à savoir un ensemble d'exercices séquentiels. Concrètement, la première partie du scénario pédagogique prévoyait le développement individuel d'un ensemble de pages web alors que la deuxième partie de la séance était consacrée à un travail collaboratif libre qui supposait d'améliorer et de réunir les différentes pages développées précédemment, au sein d'un petit « site web ». Chaque exercice était suivi de la réalisation d'un compte-rendu à rendre à la fin de la séance.

Toutes les ressources nécessaires (supports de cours, images sources) ainsi que des outils de communication (webmail, messagerie instantanée) étaient disponibles sur la plate-forme d'expérimentation. Les étudiants ont été prévenus de l'observation. Ils avaient pour consigne de ne pas communiquer autrement que par ordinateurs interposés. Présent dans la salle et ayant à sa disposition une « station d'analyse » décrite dans la suite de l'article, l'enseignant pouvait observer le déroulement de la séance d'apprentissage.

2.2. Scénario possible d'observation

Au début de la séance d'apprentissage l'enseignant souhaite être informé dès qu'un étudiant finit un exercice ou dès qu'un étudiant se trouve en difficulté

(dépassement du délai prévu pour l'exercice). En cours de séance, après avoir constaté que plusieurs étudiants se trouvent en difficulté sur le premier exercice, il décide d'observer plus précisément pour savoir si les étudiants en difficulté ont au moins fini la première partie de l'exercice. Il estime alors qu'il doit fournir à tous les étudiants un document contenant des explications supplémentaires sur le sujet. Afin de vérifier l'impact de son action, il demande ensuite d'être informé du nombre d'étudiants ayant réussi l'exercice après avoir accédé au document fourni.

Dans la deuxième partie de la séance, dédiée à une activité collaborative libre, le seul objectif d'observation de l'enseignant consiste à s'intéresser à la quantité et la nature des interactions entre les étudiants (principalement via l'échange des messages).

Afin d'avoir une telle flexibilité dans l'observation d'une séance pédagogique, une architecture logicielle particulière doit être mise en place comme on va le voir dans ce qui suit.

3. L'approche agent pour l'observation

3.1. Justification

Durant une séance pédagogique l'enseignant a une multitude de tâches à faire (contrôler la bonne compréhension du sujet, la vitesse d'avancement des apprenants, répondre à leurs appels, proposer de nouvelles activités ; etc.) ce qui l'empêche d'observer continuellement et parfaitement l'activité de chaque apprenant. Des outils d'observation disponibles sur la plate-forme d'un EIAH peuvent lui fournir une aide qu'il n'a pas à sa disposition directement dans la classe (e.g. détecter le non-respect d'une consigne) lui permettant d'adapter son comportement durant l'activité pédagogique.

Cependant, le traitement et l'analyse de la grande quantité des traces collectées durant une expérience d'observation est généralement un processus gourmand du point de vue du temps de calcul. En outre, afin de pouvoir réagir pendant l'activité pédagogique, une situation doit être observée aussitôt que possible. Le grand volume des traces distribuées parmi les stations de travail des apprenants ainsi que le besoin de déléguer et d'automatiser une partie du processus d'observation, nous a conduits à nous orienter vers une approche agent.

Les raisons qui justifient cette approche agent sont multiples [CARRON et al. 06]. Premièrement, le problème est distribué géographiquement, fonctionnellement et ne possède pas de solution « a priori » (le comportement de l'étudiant est inconnu). Ensuite, chaque station doit être autonome sans perturber l'activité de l'étudiant mais capable de communiquer avec les autres. Enfin, l'intégralité des traces hétérogènes collectées sur les stations des apprenants n'ont pas systématiquement besoin et parfois ne peuvent pas être transportées sur une autre station spécialisée pour traitement.

3.2. Des agents dédiés

Dans le cadre de l'observation de l'activité pédagogique, nous avons identifié plusieurs types d'agents correspondants aux micro-tâches d'observation que l'enseignant pourrait assurer :

Observer les actions d'un apprenant : **agent collecteur** des traces. Cet agent est chargé de collecter des traces numériques à partir des actions de l'apprenant durant l'activité pédagogique. La séquence des traces collectées est systématiquement enregistrée dans des fichiers de log sur la station de travail. L'agent est également habilité à choisir les traces collectées en fonction de ce qui lui est demandé.

Effet mémoire ou « que fait cet apprenant depuis le début du cours ? » : **agent historique**. Cet agent est chargé de chercher et d'identifier de traces dans l' « historique des actions » (l'ensemble des fichiers de log) qui se trouve sur la station de travail. Il permettra éventuellement de retrouver des informations lorsque l'on souhaite affiner ou préciser une observation.

Raisonnement ou «quel but a cette suite de micro-actions? » : **agent structuration.** Les traces collectées peuvent avoir différentes formes, peuvent être très détaillées, redondantes ou incomplètes. Ainsi l'agent structuration est spécialisé pour les regrouper, reformater ou annoter afin de les rendre plus adaptées pour l'interprétation (les amener au bon niveau d'abstraction). Cet agent est alors dédié à l'identification de patterns (ou de motifs) dans l'ensemble des traces collectées et à l'application de règles de réécriture pour changer le niveau d'abstraction des traces.

Recueil d'informations factuelles, collectives : **agent statistique**. Cet agent est doté de fonctionnalités de calcul. Il est chargé de compter et de totaliser le nombre des apprenants qui se trouvent dans un état d'avancement initialement prévu. Cet agent est évidemment informé par les autres agents (généralement par l'agent structuration) de l'état d'avancement de chaque apprenant.

Représentation de l'activité de la classe à un instant donné: **agent visualisation.** Cet agent est spécialisé dans la visualisation/présentation des traces ainsi que dans l'affichage de résultats ou messages (autres que la trace). Il permet ainsi de suivre plus aisément ce qu'il se passe au cours de l'activité pédagogique.

3.3. Déploiement des agents et mise en place dans le cadre de l'expérimentation

Le déploiement des agents sur les machines est lié à l'objectif d'observation visé pour la séance pédagogique. L'approche permet de déterminer les agents nécessaires durant la séance ainsi que l'endroit où il faut déployer ces agents.

Nous pouvons alors noter que certains agents ne sont pas obligatoires : un agent peut être absent ou inactif (e.g. l'agent collecteur inactif sur une station dédiée à l'analyse comme celle de l'enseignant par exemple). Il est également possible d'avoir plusieurs agents d'un même type sur la même station (e.g. plusieurs agents de visualisation sur la station d'analyse).

En fonction du rôle de l'utilisateur de la station dans l'activité pédagogique, nous pouvons imaginer plusieurs scénarios de déploiement du système multi-agents et identifier ainsi plusieurs types de station. Par exemple, on peut définir qu'une « station d'analyse » sera composée d'agents structuration, statistique et visualisation, qu'une station observée ou « station de travail » sera généralement équipée des agents collecteur, historique et structuration, qu'une « station visualisation » ne comportera que des agents de visualisation. On peut même définir une « station multi-rôles » qui embarquera l'ensemble de tous les agents, etc. L'avantage est de pouvoir adapter la puissance ou les ressources de la machine en fonction de son utilisation : il est possible par exemple, de délocaliser uniquement la visualisation (nécessitant peu de ressources de calcul ou d'espace mémoire) sur un périphérique mobile comme un assistant personnel (PDA) afin de pouvoir aisément se déplacer entre différentes salles. Le principe du système multi-agents nous apporte cette flexibilité.

Dans le cadre de cette expérimentation (cf. section 2), nous avons choisi de ne pas offrir aux apprenants de fonctions d'observation sur leur propre travail (nous ne nous sommes pas intéressés à la meta-cognition dans cette expérimentation). Par conséquent, nous avons prévu pour les apprenants des « stations de travail» et pour l'enseignant une « station d'analyse » comme décrites ci-dessus.

4. Nécessité d'un langage de configuration agent adapté à l'observation

4.1. Compétences et besoins

Les agents décrits dans la section précédente ont des compétences mais aussi des besoins spécifiques (des informations ou résultats qu'ils ne savent pas obtenir directement). Ainsi, l'agent collecteur ou l'agent historique ont besoin de la description des traces à sélectionner parmi les traces collectées ou enregistrées dans les fichiers de log. L'agent structuration applique ses règles de réécriture de traces par l'identification de patterns qui lui ont été fournis. Enfin, le fonctionnement de l'agent statistique nécessite la liste des étapes du parcours pédagogique à surveiller (fin d'un exercice, consultation d'une aide).

Les notions de compétences et de besoins spécifiques imposent aux agents de coopérer pour résoudre un but général d'observation. Par exemple, les traces collectées et sélectionnées par les agents collecteurs sont regroupées via un agent structuration, le résultat étant affiché à l'analyste grâce à un agent visualisation. Afin de pouvoir coopérer chaque agent doit déclarer ses compétences : dans notre expérimentation, nous avons considéré que les compétences agent étaient associées à un type d'agent (cf. section 3.2).

Enfin, les agents peuvent coopérer mais également réagir de manière autonome. La réaction d'un agent est conditionnée par l'apparition d'une situation initialement prévue et peut ensuite engager l'agent dans une nouvelle coopération.

4.2. Réactions d'un agent

Un agent lancé dans une tâche d'observation doit pouvoir réagir dans plusieurs situations : identification d'un pattern dans l'ensemble des traces collectées, dépassement d'un délai prévu pour observer, nombre limite d'apprenants accédant à une étape du parcours atteint. La réaction d'un agent est composée d'une suite d'actions exécutables par lui-même ou qui peuvent être adressées à un autre agent ou bien à un ensemble d'agents (on peut ainsi reconfigurer tous les agents ayant un type spécifié).

Dans une approche d'observation agent, tous ces aspects doivent pouvoir être représentés formellement dans un langage de configuration agent. De plus, afin d'être adapté à l'observation, le langage (manipulable aussi par un agent humain) doit être suffisamment flexible pour permettre la (re)configuration dynamique des agents car l'objectif d'observation est généralement ajusté dynamiquement durant la session d'apprentissage.

Les langages de contenu agent existants (notamment fipa-sl [FIPA-SL 07] et fipa-kif [FIPA-KIF 07]) ne sont pas adaptés pour représenter une tâche d'observation telle que décrite ci-dessus en mettant en évidence la réaction des agents. Bien que ces langages permettent de décrire généralement des assertions, ils ne fournissent pas un cadre précis pour décrire *quand* et *comment* un agent doit réagir durant un processus d'observation.

Dans la section suivante, nous exposons notre proposition pour un langage de configuration agent adapté à l'observation. Notre langage permet de décrire une tâche d'observation spécifique pour chaque type d'agent en décrivant également la réaction attendue de l'agent.

5. Un langage de configuration agent adapté à l'observation

5.1. Description du langage

L'élément de base de notre langage de configuration agent est la règle. Une règle est représentée par une construction du type « condition /action » appliquée dans une structure de contrôle répétitive (un « do-od ») basée sur les commandes gardées de Dijkstra [DIJKSTRA 75]. On associe alors à chaque agent du système multi-agents un ensemble de règles qui s'exécutent de manière répétée dans une boucle infinie. Une règle implique la description de sa partie gauche (la garde ou la condition) et de sa partie droite (l'action associée).

La partie gauche de la règle décrit la tâche d'observation. Une règle peut décrire un pattern à l'intérieur des traces d'activité permettant d'identifier celles qui se révèlent intéressantes (cf. agent collecteur) ou décrire une séquence de traces en vue de structuration (cf. agent structuration). La règle peut également décrire à ce stade

des contraintes sur le domaine (i.e. une date, un nombre d'étudiants réalisant une telle activité).

La partie droite de la règle décrit la réaction de l'agent précisée par un ensemble d'actions (e.g. un message pour un autre agent ; une reconfiguration d'agent ; le lancement d'une application). La réaction de l'agent peut se déclencher principalement au moment de : *a)* l'identification du pattern décrit dans la partie gauche de la règle, dans l'ensemble de traces collectées ; *b)* le dépassement d'un délai temporel prévu pour l'apparition de la situation en cours d'observation (spécifique pour tous les agents sauf l'agent historique).

Le comportement d'un agent est ainsi décrit avec un ensemble de règles. L'ensemble des règles est modifiable par l'ajout/la suppression de règles dynamiquement durant le processus d'observation. Le langage fournit les moyens de gérer l'ensemble de règles associé à chaque agent : l'activation ou la désactivation d'une règle, la suppression d'une règle ou de toutes les règles connues par l'agent.

5.2. Grammaire du langage

Nous présentons dans ce qui suit, une partie de la grammaire (simplifiée par manque de place) qui permet de décrire une règle. Dans notre description BNF, les éléments dont le nom commence par « id » représentent une séquence alphanumérique quelconque sans espaces qui décrit un nom. L'élément « date » décrit une séquence date-time, « regex » une expression régulière et « address » une construction du type « ip::port » qui permet de localiser un agent dans le système.

```
<rule> ::= "(" "defrule" idRule [<state>][<global>][<pattern>]
                    [<step>][<delay>][<not-found>] -> [<reaction>] ")"
<state> ::= "(" "state" ("enabled" | "disabled") ")"
<global> ::= "(" "global" <expression>+ ")"
<pattern> ::= "(" "pattern" <track>+ ")"
<track> ::= "(" "track" <expression>+ ")"
<expression> ::= "{" idAttribute <operator> ["distinct"] <value> "}"
<operator> ::= "==" | "!=" | ">" | "<"
<value> ::= “"” (regex | date) “"”
<step> ::= "(" "step" “"” idStep “"” (0|[1-9][0-9]*) ")"
<delay> ::= "(" "delay" ">" [1-9][0-9]* -> <reaction> ")"
<not-found> ::= "(" "not-found" -> <reaction> ")"
<reaction> ::= <action>*
<action> ::= "(" "action" <agent-identifier> idMethod <args> ")"
<agent-identifier> ::= "(" "agent-identifier"
          [":name" idAgent] [":address" address] [":type" idType] ")"
<args> ::= "(" "args" ([":arg"[0-9]] “"” [^\r\n\"\\]* “"”)+ ")"
```

Dans la partie droite de la règle, il est possible de faire appel à des séquences de caractères identifiées dans l'ensemble des traces collectées en utilisant des expressions régulières [REGEX 07]. Nous présentons à présent, un exemple de (ré)action d'un agent (❷) qui utilise le nom d'un fichier identifié (❶) dans la trace :

```
❶ (track {action == "open file : (.+).htm"})
❷ (action (agent-identifier :name toto) show
            (args :arg0 “file $1 opened !”))
```

De plus, tous les arguments d'une action décrite dans un contexte de réaction agent peuvent appeler les observés (valeur d'*idAttribute*) décrits par le pattern de la règle, en utilisant l'identifiant de l'observé précédé par le caractère « & » :

```
(args :arg0 "&date : &user début exercice2")
```

Nous présentons ci-dessous la grammaire de commandes qui permettent de gérer l'ensemble des règles associées à un agent: la suppression et le changement de l'état des règles (activation / désactivation des règles).

```
<undef> ::= "(" "undef" (idRule | "all") ")"
<change-state> ::= "(" ("enable" | "disable") idRule ")"
```

5.3. Quelques règles pour le fonctionnement de l'expérimentation

Dans ce paragraphe, nous présentons un exemple de configuration des agents en utilisant ce langage de configuration pour l'expérimentation décrite en section 2. L'objectif de cet exemple est de mettre en lumière la flexibilité de l'approche qui permet de définir et de reconsidérer à tout moment le comportement d'un agent ou d'un groupe d'agents durant le processus d'observation de l'activité pédagogique.

5.3.1. Mise en place de règles de comportement pour l'expérimentation

Nous avons défini au début de l'expérimentation, une règle pour chaque agent structuration déployé sur les stations de travail des étudiants. Cette règle a permis aux agents de repérer la fin du premier exercice (l'élaboration des pages web *index.php* et *exercice1.php*). La réaction de la règle prévoit l'envoi de messages de notification à l'enseignant en cas de réussite et/ou de constat d'étudiant en difficulté (dépassement du délai de réalisation prévu ; dans notre exemple : 15 minutes).

```
(defrule rule-st1
  (pattern
     (track {eventValue == ".*index.php.*"} {focus == "jEdit.*"})
     (track {eventValue == ".*exercice1.php.*"} {focus == "jEdit.*"})
     (track {eventValue == ".*compterendu.odt.*"}))
  (delay > 900000 ->
     (action (agent-identifier :name prof :type professeur) show
       (args :arg0 "&date : &user en retard sur l'ex.1 !")))
  ->
  (action (agent-identifier :name prof :type professeur) show
       (args :arg0 "&date : &user a fini l'ex. 1!")))
```

5.3.2. Raffinement de l'objectif d'observation

Un étudiant qui n'a pas commencé la deuxième partie de l'exercice (après le dépassement du délai prévu) a été considéré en difficulté sur la première partie de l'exercice. Pour obtenir cette information, nous avons utilisé les agents historiques résidant sur les stations de travail des étudiants.

Un document contenant des explications supplémentaires est alors mis à disposition des étudiants. Afin de valider l'utilité de ce document, nous avons décrit une nouvelle règle (*rule-h2*) pour les agents historiques localisés sur les stations de

travail. Cette règle a permis de faire le lien entre la consultation du document fourni et la réussite du reste de l'exercice. Dans notre expérimentation seuls 44,4% des étudiants ont consulté le document fourni (la plupart étaient en difficulté sur la première partie de l'exercice). 87.5% d'entre eux ont fini l'exercice dans les 10 minutes après la consultation du document.

```
(defrule rule-h2
   (pattern
      (track {eventValue == ".*aide1.pdf.*"} {focus == ".*Reader.*"})
      (track {eventValue == ".*compterendu.odt.*"}))
   ->
   (action (agent-identifier :name prof :type professeur) show
      (args "&date : &user fin après la consultation de l'aide !")))
```

La deuxième partie de la séance, assez différente pédagogiquement puisqu'elle repose sur le travail collaboratif, nécessite la reconfiguration dynamique des agents pour observer l'activité de manière plus adéquate.

5.3.3. Reconfiguration complète des agents

La reconfiguration des agents pour un nouveau type d'observation nécessite la mise à jour de l'ensemble des règles agent, processus qui peut s'exécuter dynamiquement par rapport à une situation observée. Dans notre expérimentation, nous avons défini une règle « inactive » pour tous les agents structuration (la règle *rule-st2*) permettant de tracer les étudiants ayant communiqué via la messagerie instantanée. La règle a été activée dynamiquement pour chaque étudiant au début de l'activité collaborative (la règle *rule-st3*).

```
(defrule rule-st2
   (state disabled)
   (pattern (track {eventValue == distinct "(.*) - Instant Message.*"}
                     {eventType == "keyKeyboard"}))
   ->
   (action (agent-identifier :name prof :type professeur) show
          (args :arg0 "&date : msn &user avec $1")))

(defrule rule-st3
   (pattern (track {eventValue == ".*exercice collaboratif.*"}))
   ->
   (action (agent-identifier :address 127.0.0.1::45678) reconfig
           (args "(enable rule-st2) (undef rule-st3)")))
```

6. Discussion

Nous avons présenté ici quelques règles qui nous ont permis d'observer un certain nombre de situations intéressantes dans le cadre de cette expérimentation. Mais l'expérimentation nous a également permis de tester une grande variété des règles agent comme par exemple des règles pour être informé du respect du scénario pédagogique, du nombre des messages échangés par apprenant ou par groupe d'apprenants, etc. Cela nous a permis de valider l'expressivité du langage dans le cadre de ce travail. Enfin, l'expérimentation s'est déroulée avec tous les acteurs dans la même salle, ce qui a permis à l'enseignant de s'assurer de la validité des résultats obtenus grâce à notre système d'observation.

Si cette présence de l'enseignant est souhaitable dans le cadre d'une première expérimentation, il nous semble qu'elle doit introduire des biais par rapport à une situation d'apprentissage à distance, par exemple. Lorsque l'enseignant est présent, le comportement des apprenants en est naturellement modifié. De même, il sera plus difficile de faire respecter les consignes de communication en situation d'échec, car si l'enseignant est à côté de lui, l'apprenant sera naturellement tenté de lui poser des questions oralement.

Dans notre expérimentation, nous avons initialement configuré les agents au début de la séance, en fonction des attentes de l'enseignant. Le système multi-agents[1] initialement configuré est autonome et donc capable d'observer et de réagir avec un minimum d'interaction de la part de l'enseignant. Cependant, dès que l'objectif d'observation s'affine, il faut inévitablement activer d'autres règles. Il est difficile d'imaginer que toutes les règles auront été prévues à l'avance. Il est donc nécessaire d'avoir en permanence un ingénieur pédagogique (le rédacteur des règles) pour rendre notre système optimal. Ceci est assez peu réaliste dans le monde éducatif, l'enseignant n'ayant pas ces habitudes de travail. D'autre part, le langage tel qu'il est actuellement défini ne permet évidemment pas à un enseignant d'écrire de nouvelles règles. Nous voyons donc ici un frein important à la mise en place de notre approche. Une simplification du langage (introduction d'actions graphiques) est une des pistes à envisager. Une autre consisterait à définir une bibliothèque de comportements types réutilisables et adaptables directement sur des agents.

7. Conclusion

Le travail présenté dans cet article propose donc une approche agent pour l'observation de l'activité pédagogique. Les agents « observateurs » logiciels sont (re)configurables grâce à un langage de configuration agent adapté à l'observation. L'approche agent apporte à l'observation de la flexibilité, permettant de choisir dynamiquement quand et quoi observer pendant l'activité pédagogique.

Dans le domaine de l'éducation, plusieurs acteurs peuvent bénéficier d'une telle approche d'observation : enseignants, apprenants (meta-cognition : avoir une vue de réflexion sur son chemin d'étude) ou chercheurs en sciences sociales.

Les perspectives à suivre sont multiples. Tout d'abord nous prévoyons d'autres agents spécialisés pour l'observation. Dans l'expérimentation présentée nous avons utilisé un agent visualisation simple qui nous a permis d'afficher les résultats obtenus textuellement. Actuellement, nous sommes en train de développer un ensemble d'agents de visualisation plus complexes qui permettront l'analyse graphique de l'activité [FRANCE et al. 06]. Nous envisageons aussi de développer des agents capables de manipuler d'autres types de traces (e.g. des images).

---

[1] Les agents sont développés en langage java et le système ne nécessite pas de plate-forme multi-agents dédiée afin de pouvoir être rapidement mis en œuvre n'importe où.

Au vu de l'expérimentation menée, le langage se révèle suffisamment expressif pour mettre en place assez simplement les mécanismes d'observation que nous souhaitons. Cependant, bien que le langage de configuration agent proposé manipule des concepts simples, nous sommes bien conscients qu'une interface intuitive d'utilisation doit être fournie à l'utilisateur pour l'élaboration de nouvelles règles dynamiquement durant la séance d'apprentissage.

Ce travail de recherche a été réalisé dans le cadre du cluster « informatique, signal, logiciels embarqués », financé par la Région Rhône-Alpes (France).

8. Bibliographie

[CARRON et al. 06] Carron T., Marty J.C., Heraud J.M., France L., « Helping the teacher to re-organize tasks in a collaborative learning activity: an agent based approach », *ICALT 2006*, Kerkrade, The Netherlands, 2006, p. 552-554.

[CASTERAN et al. 00] Casteran J.C., Gleizes M.P., Glize P., « Des methodologies orientées multi-agents », *JFIADSMA'00*, Saint-Jean la Vêtre, Loire, France, p.191-207.

[CHAMPIN et al. 04] Champin P.A., Prié Y., Mille A., « MUSETTE: a Framework for Knowledge Capture from Experience», *EGC'04*, Clermont Ferrand, France, 2004.

[COURTIN & TALBOT 06] Courtin C., Talbot S., « Trace Analysis in Instrumented Collaborative Learning Environments », *ICALT 2006*, Kerkrade, The Netherlands, 2006.

[DIJKSTRA 75] Dijkstra E.W., « Guarded Commands, Nondeterminacy and Formal Derivation of Programs », *Communications of the ACM*, vol 18, n° 8, 1975, p. 453-457.

[IKSAL & CHOQUET 05] Iksal S., Choquet C., « An Open Architecture for Usage Analysis in a E-Learning Context ». *ICALT 2005*, Kaohsiung ,Taiwan, 2005, p. 177-181.

[FRANCE et al. 06] France L., Heraud J.M., Marty J.C., Carron T., Heili J., « Monitoring virtual classroom: Visualisation techniques to observe student activities in an e-learning system », *ICALT 2006*, Kerkrade, The Netherlands, 2006, p. 716-720.

[MARTY et al. 07] Marty J.C., Heraud J.M., Carron T., France L., « Matching the Performed Activity on an Educational Platform with a Recommended Pedagogical Scenario: a Multi Source Approach », *Journal of Interactive Learning Research*, vol 18, n° 2, 2007.

[SETTOUTI et al. 06] Settouti L.S., Prié Y., Mille A., Marty J.C., « Système à base de trace pour l'apprentissage humain », *TICE 2006*, Toulouse, France, 2006.

9. Références sur le WEB

[FIPA-SL 07] FIPA SL (Semantic Language) Language Specification, http://www.fipa.org/specs/fipa00008/, janvier 2007.

[FIPA-KIF 07] FIPA KIF (Knowledge Interchange Format) Language Specification, http://www.fipa.org/specs/fipa00010/, janvier 2007.

[REGEX 07] Regular Expressions, http://www.regular-expressions.info, janvier 2007.