

# Vers un agent dialogique proactif

F. Delecroix

Fabien.Delecroix@etudiant.univ-lille1.fr

M. Morge

Maxime.Morge@univ-lille1.fr

J.-C. Routier

Jean-Christophe.Routier@univ-lille1.fr

Laboratoire d'Informatique Fondamentale de Lille,  
Université Lille 1, France

## Résumé

*La plupart des agents conversationnels déployés sur les sites de vente en ligne se cantonnent au rôle de foire aux questions interactive. Le retour d'expérimentation nous apprend que les attentes des utilisateurs vis-à-vis de ces agents conversationnels ne sont pas comblées. Dans cet article, nous définissons un agent proactif expert destiné à mieux répondre à ce type d'application. A partir d'un scénario, nous identifions les caractéristiques que doit exhiber un tel agent dialogique proactif : prise d'initiative, adaptabilité, acquisition d'information, etc. Notre implémentation s'appuie sur le langage AIML initialement conçu pour des agents purement réactifs. Nous en identifions le potentiel et les limites pour notre approche.*

**Mots-clés :** Agents conversationnels, Modèles de comportements, Systèmes mixtes

## Abstract

*Most of the conversational agents which are deployed on e-commerce website take the place of FAQ. Their usage by stakeholders is deceptive. In this paper, we propose a proactive dialogical agent for this kind of application. Firstly, we identify with a walk-through example the agent overt properties : leadership, adaptability, information seeking, etc. Our implementation is based upon the AIML language which has been proposed for purely reactive agents. We identify here the possibilities of this language and its limits for our approach.*

**Keywords:** Conversational agents, Behavioral models, Mixed systems

## 1 Introduction

Les agents conversationnels sont souvent employés pour fournir de l'assistance à un agent humain. Leur attrait principal est de proposer une interface naturelle dans la langue de l'utilisateur. Les agents conversationnels assistants font l'objet de nombreux travaux [12, 17, 11, 7, 8, 13, 16]. Dans la majorité des cas, ils consti-

tuent un interlocuteur pour répondre aux questions que se pose l'utilisateur. Leur rôle se résume alors à celui de foire aux questions (FAQ) interactive [1].

Pour rendre l'interaction encore plus intuitive, la plupart de ces agents sont incarnés par des avatars. On parle d'agents conversationnels animés (ACAs). La personnalisation d'un interlocuteur crédibilise davantage le système, Lester et *al.* parlent de *Persona Effect* [12]. On trouve des ACAs de plus en plus sophistiqués [15], exprimant même des émotions [16]. Cet état de fait vient renforcer des attentes déjà fortes induites par la formulation libre. Cependant, l'utilisateur est souvent frustré par les compétences (métiers et linguistiques) limitées des agents conversationnels [14].

Notre approche consiste à envisager l'agent conversationnel comme un conseiller, expert métier, apte à aider l'utilisateur dans ses prises de décisions [13]. L'agent conversationnel revêt alors un caractère proactif en menant le dialogue et en agissant dans le but de répondre aux besoins de l'utilisateur. C'est dans la perspective d'élaborer un tel agent que s'inscrit cet article.

Dans la section 2, nous définissons la problématique applicative et nous proposons un scénario concret y correspondant. Dans la section 3, nous identifions à partir de ce scénario les caractéristiques que doit exhiber l'agent dans un tel dialogue. Nous proposons une implémentation à l'aide du langage AIML (section 4) et nous situons les limites du langage dans ce cadre. Enfin, nous concluons l'article en dressant quelques perspectives.

## 2 Problématique applicative

Nous nous intéressons au cas particulier de la vente en ligne. Dans ce cadre, l'objectif est d'améliorer le taux de transformation (le ratio du nombre de ventes sur le nombre de visites) ainsi que la satisfaction client. Classiquement, la recherche d'un produit dans de tels sites se fait via un moteur de recherche, éventuellement accompagné d'une série de filtres.

Cette approche suppose non seulement que le client connaisse précisément le produit recherché mais également le moteur lui-même. Utiliser un agent conversationnel permet de s'abstraire de ces hypothèses fortes. La mission de l'agent consiste alors à cibler au mieux les besoins de l'utilisateur, proposer des produits adéquats et répondre aux interrogations éventuelles. Autrement dit, l'agent joue un rôle similaire à celui d'un conseiller-vendeur dans un magasin. Il met au service du client sa connaissance du catalogue de produits et son expertise métier pour lier des produits à des besoins.

Nous pouvons situer ce type d'échange sur la grille d'analyse des dialogues (cf. Table 1) proposée par Walton et Krabbe [18]. Les six catégories de dialogues proposées se distinguent de par leur situation initiale et leur but principal.

Nous considérons que la conversation entre l'utilisateur et l'agent assistant est une instantiation de deux dialogues de demande d'information. Un dialogue de demande d'information intervient lorsqu'un des participants souhaite disposer d'une connaissance qu'il suppose être détenue par son interlocuteur. Ici,

- l'utilisateur a des besoins mais ignore l'information sur les produits ;
- l'agent dispose de l'information sur les produits mais ignore quels sont les besoins de l'utilisateur.

Dans la conversation, le but commun des deux participants est d'identifier au moins un produit répondant aux besoins de l'utilisateur. C'est ce but que va chercher à atteindre notre agent dialogique proactif, en anglais *Proactive Dialogical Agent* (PDA).

Dans le dialogue représenté dans la figure 1, le client souhaite acheter un vélo. Cependant, il n'a pas identifié précisément le produit (i.e. toutes les caractéristiques) qui correspond à son besoin (i.e. un vélo pour sa fille). A cette intention, l'agent le guide dans sa recherche. Cet exemple sert d'illustration pour l'ensemble de cet article.

### 3 Caractéristiques

Dans cette section, nous allons identifier les caractéristiques que doit exhiber notre PDA.

Nous considérons ici que l'interaction entre un PDA et un utilisateur est uniquement textuelle. Dans notre exemple, le dialogue se déroule via un navigateur web.

Comme représenté dans la Figure 2, le dialogue est découpé en tours de parole. Se plaçant du point de vue de l'agent, on appellera **entrée** un énoncé produit par l'utilisateur et **sortie** un énoncé produit par l'agent. Le dialogue alterne donc sortie et entrée.

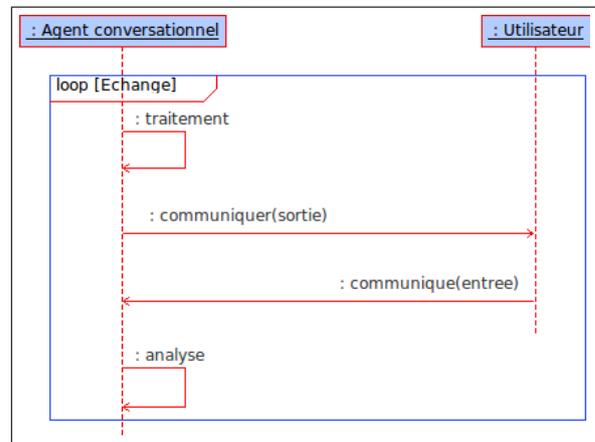


FIGURE 2 – Diagramme de séquence entre le PDA et l'utilisateur

#### 3.1 Initiative

Afin de construire un PDA qui mène la conversation, il faut qu'il puisse être à l'initiative de celle-ci. Un agent proactif ne doit pas rester passif et se tenir simplement à disposition. En effet, le PDA doit **engager le dialogue**, plutôt que d'attendre une requête de l'utilisateur.

Plusieurs éléments déclencheurs peuvent conduire le PDA à entamer le dialogue. Le PDA doit être muni d'un mécanisme de décision qui prend la forme de règles de déclenchement (en anglais, *triggering rules*). Les informations sur le parcours de navigation de l'internaute ainsi que les temps de flottement sont des éléments pertinents pour déterminer la politique d'apparition de l'agent sans qu'il soit trop intrusif [9].

Dans notre exemple (cf Fig. 1), c'est le parcours de navigation dans le rayon vélo qui déclenche l'intervention du PDA (énoncé 1). De même, la pause prise par l'utilisateur après la proposition de produit déclenche la relance du dialogue par le PDA pour conclure l'échange (énoncé 18). On peut noter que le *timeout* permet de passer au tour de parole suivant.

#### 3.2 Réactivité

Notre approche est complémentaire à celle adoptée par les agents réactifs. La proactivité de

Situation initiale → But principal ↓	Conflit	Problème ouvert	Ignorance d'un participant
Accord stable i.e. Résolution	persuasion	enquête	demande d'information
Règlement pratique i.e. Décision (ou non) d'agir	négociation	délibération	∅
Arrangement provisoire	éristique	∅	∅

TABLE 1 – Aperçu systémique des types de dialogues [18]

*Navigation dans le rayon vélo*

1. **PDA** : Bonjour. Je suis conseiller vélo. Puis-je vous aider ?
2. **Client** : Oui, je cherche un vélo pour ma fille.
3. **PDA** : Quel âge a-t-elle ?
4. **Client** : 5 ans.
5. **PDA** : Vous souhaitez un 14" ou un 16" ?
6. **Client** : Je ne sais pas.
7. **PDA** : Combien mesure votre enfant ?
8. **Client** : 1m05 environ.
9. **PDA** : Je vous conseille plutôt le 14". Quel est votre budget ?
10. **Client** : Pas trop cher.
11. **PDA** : Je n'ai pas compris. Donnez-moi un prix.
12. **Client** : Environ 80.

*Affichage des produits : 14" AQUA, 14" ELEPHAN, 14" TICHA*

13. **PDA** : Voici les produits correspondants. Avez-vous une préférence sur la couleur ?
14. **Client** : Ont-ils des roulettes ?
15. **PDA** : Oui, ces vélos ont des stabilisateurs. Quelle couleur désirez vous ?
16. **Client** : Rose.

*Affichage d'un produit : 14" TICHA, rose, 84.90 €*

17. **PDA** : Vous convient-il ?

*Timeout*

18. **PDA** : La garantie est de 5 ans cadre et fourche - 2 ans pièces. Vous le prenez ?
19. **Client** : Finalement je vais prendre le 14" AQUA.
20. **PDA** : D'accord, j'ajoute dans le panier le 14" AQUA.
21. **Client** : Peut-on payer par carte bleue ?
22. **PDA** : Le paiement par carte bleue est possible.

*Affichage page paiement CB*

FIGURE 1 – Conversation entre un agent vendeur proactif et un client

l'agent ne vient pas se substituer à sa réactivité mais s'y additionner.

Comme indiqué dans la figure 2, le bot doit **analyser le contexte dialogique**, en particulier l'entrée précédente, afin de **le traiter pour générer sa sortie**. Dans le cas particulier de la première sortie, le dialogue est amorcé par un énoncé initial pré-établi.

Dans notre exemple, les sorties du PDA peuvent prendre la forme d'effet de bord, par exemple l'affichage d'une page web (énoncé 13). Même si ce n'est pas sa vocation première, un PDA doit être capable de répondre aux requêtes du client (énoncé 22) comme le font les agents purement réactifs.

### 3.3 Adopter des buts

Au fur et à mesure de la conversation, le PDA **adopte les buts de l'utilisateur** pour répondre à ses besoins.

Dans un premier temps, le PDA adopte un but initial, i.e. identifier un produit. Pour ce faire, le PDA doit résoudre des sous-buts consistant à déterminer les attributs du produit adéquat.

Dans notre exemple, le PDA adopte le but initial « trouver vélo » pour répondre à la requête de l'utilisateur (énoncé 2). Pour le résoudre, le PDA doit identifier le produit (14" AQUA) en fonction des attributs (14", blanc, 84,90 €). Simultanément, le PDA souhaite raffiner les besoins exprimés et déterminer l'attribut du produit pour chaque critère (taille, prix, couleur).

À tout moment de la conversation, les énoncés de l'utilisateur sont susceptibles de déclencher de nouveaux buts. Ils viennent s'ajouter à ceux déjà adoptés, on a alors plusieurs buts à résoudre en concurrence.

### 3.4 Orienter le dialogue

Un PDA est capable de diriger l'échange, d'**imposer les sujets** et le contexte dialogique.

Contrairement à l'approche *chatbot*, la majorité des sorties du PDA sont des questions et la majorité des entrées du PDA sont des réponses à ces questions. De cette manière, le PDA impose le contexte dialogique et réduit le champ sémantique pour l'interprétation des entrées.

Dans notre exemple, le PDA pose des questions (énoncés 3, 5, 7, 9, 11, 13, 15, 17 et 18) et l'uti-

lisateur y répond (énoncés 4, 6, 8, 10, 12, 16, 19).

À chacun de ses tours de parole, notre agent a la possibilité de définir le sujet sur lequel portera la sortie qu'il va produire et donc d'orienter le dialogue. Ce choix est fait en fonction de ses buts actuels et de son avancement dans ceux-ci. Dans le cas présent, le besoin de l'utilisateur correspond à un vélo pour sa fille (énoncé 2). Le PDA demande à l'utilisateur de préciser son besoin pour y faire correspondre un produit. Il le questionne par exemple sur des critères comme la taille du vélo (énoncé 5) ou encore son prix (énoncé 9).

### 3.5 Acquérir de l'information

Comme indiqué dans la section 3.3, le but du PDA est d'assister l'utilisateur. Il doit pour cela **collecter des données** pour réaliser cette tâche.

La proactivité du PDA est une condition *sine qua non* de l'acquisition de cette information. En effet, étant donné le besoin exprimé par l'utilisateur, le PDA oriente le dialogue selon ses sous-buts (cf. section 3.3) afin de le préciser.

Le PDA doit ensuite extraire l'information utile des entrées-réponses de l'utilisateur. L'analyse syntaxico-sémantique est alors facilitée par l'orientation imposée par l'agent (cf. section 3.4).

Par exemple, dans notre scénario, l'énoncé « environ 80 » (énoncé 12) est interprétable par le PDA comme un prix en raison du contexte dialogique défini par la demande précédente (i.e. énoncé 11).

L'acquisition d'information permet au PDA la mise à jour de ses connaissances quant aux besoins de l'utilisateur. La réponse de l'utilisateur peut amener à résoudre un sous-but (énoncé 16) ou à en créer un nouveau (énoncé 6) provoquant une nouvelle demande d'information. Le cheminement du PDA est ainsi proche de celui d'un système expert par chaînage mixte : le besoin de résoudre un but conduit à chercher l'information manquante (par chaînage arrière) et à interroger l'utilisateur sur celle-ci. L'information fournie permet à son tour la déduction (par chaînage avant) de nouvelles informations et le cycle se répète jusqu'à satisfaction du but, ou constat d'échec.

### 3.6 Adaptabilité

Dans le dialogue, le client n'est pas toujours à même de répondre aux questions posées par l'agent. C'est alors au PDA de s'adapter de manière proactive, i.e. d'**ajuster son comportement au profil de l'utilisateur**.

Dans le cas particulier de la vente, il s'adapte en collectant des informations sur l'usage plutôt que sur les besoins. De par son expertise, il déduira ensuite les caractéristiques que doit posséder le produit.

Dans notre exemple, l'utilisateur ne sait pas quelle taille de vélo est la plus appropriée (énoncé 6). L'âge et la taille de la fille sont des informations pertinentes pour déduire une taille de vélo adéquate. Notre PDA oriente donc le dialogue afin d'acquérir ces informations.

### 3.7 Robustesse

Le traitement du langage naturel permet d'aller au-delà d'un simple formulaire au travers duquel l'utilisateur répond à des questions fermées. Il n'est cependant pas possible pour le concepteur du PDA de prévoir l'intégralité des réponses possibles. Comme pour la conception d'agents conversationnels réactifs, il est nécessaire de **traiter les situations d'incompréhension** en demandant à l'utilisateur de reformuler son énoncé.

Dans notre exemple, le PDA n'est pas capable d'interpréter l'énoncé 10 de l'utilisateur. Il demande donc de reformuler cet énoncé de manière plus précise. *A contrario*, bien que l'énoncé 14 ne réponde pas à la question posée mais porte sur un autre sujet, le PDA fait preuve de flexibilité afin de l'interpréter.

### 3.8 Flexibilité

Le PDA est muni d'une **autonomie ajustable**. En effet, dans le cas idéal, l'utilisateur suit l'orientation du dialogue donnée par l'agent tout au long de la conversation. C'est alors lui qui est dans une attitude réactive. Cependant, il est possible que l'utilisateur souhaite réorienter le dialogue. En conséquence, sa réponse ne correspond pas à la question posée par l'agent. Le PDA doit alors s'adapter à cette situation d'échappement puis éventuellement poursuivre le dialogue initial.

Dans notre exemple, après que le PDA lui ait demandé ses préférences sur la couleur (énoncé

13), le client ne répond pas et souhaite savoir si les vélos ont des roulettes (énoncé 14). Cette entrée n'a pas de sens dans le contexte dialogique du PDA. Il lui faut donc l'interpréter à un autre niveau, y répondre de manière purement réactive et recadrer la conversation (énoncé 15).

### 3.9 Conclusion

Nous venons d'identifier les caractéristiques constituant les lignes directrices pour la conception d'un agent dialogique proactif. Elles définissent le comportement de l'agent et lui permettent de couvrir un large champ dialogique. Avec de tels caractéristiques, notre PDA peut mener la conversation et ainsi acquérir l'information requise afin de proposer des solutions pertinentes à l'utilisateur.

D'autres caractéristiques complémentaires plus classiques sont envisageables pour améliorer le PDA. Parmi celles-ci, la fiabilité est la capacité de l'agent à bien interpréter les énoncés de l'utilisateur. On peut dans cette optique mettre en place un mécanisme de validation successif à l'acquisition d'une connaissance. En énonçant clairement l'information assimilée par le bot, on réduit l'impact d'éventuelles erreurs. Un autre moyen de renforcer la crédibilité de l'agent consiste à diversifier les énoncés qu'il produit. Pour un agent proactif, il s'agit notamment d'éviter la répétition d'une question sous la même forme (énoncés 13 et 15).

## 4 AIML

Après avoir établi les différentes caractéristiques d'un PDA, nous étudions la faisabilité de chacune d'entre elles en AIML (Artificial Intelligence Markup Language). Dans cette section nous présentons le langage AIML et l'implémentation que nous avons réalisée à l'aide de celui-ci.

### 4.1 Langage AIML

AIML [2] est un langage dérivé du XML destiné à la programmation d'agents clavardeurs (en anglais, *chatterbots*). Il a notamment été utilisé pour le développement d'agents conversationnels de type ELIZA [10], enrichis et étendus, tels que A.L.I.C.E. [3], agent ayant remporté plusieurs fois le prix Loebner [6].

Le principe d'un *bot* conçu en AIML consiste à filtrer les entrées reçues pour générer ses sorties.

```

1 <category>
2   <pattern>* CARTE BLEUE</pattern>
3   <that>_</that>
4   <template>
5     Le paiement par carte bleue est
6     possible.
7     <srai>TROUVER</srai>
8   </template>
9 </category>

```

FIGURE 3 – Exemple de catégorie AIML

En AIML, une règle dialogique (cf. Fig. 3) s'exprime sous la forme d'une `category` associant un traitement (`template`) à un motif donné (`pattern`), par convention en capitales.

Les éléments optionnels `topic` et `that` permettent d'implémenter dans une certaine mesure la notion de contexte dialogique. Un `topic` regroupe plusieurs catégories sous un même motif. L'élément `that` contient un motif filtrant la dernière sortie produite par le bot. Les motifs peuvent être composés de mots et/ou de jokers. Il existe uniquement deux jokers en AIML «`*`» et «`_`» qui se substituent à un ou plusieurs mots quelconques. Ils se distinguent par leur priorité dans l'algorithme de filtrage. Par défaut, `topic` et `that` valent «`*`».

On obtient alors trois niveaux dans l'algorithme de filtrage d'AIML. Ainsi, dans notre exemple Fig. 3, pour que le `template` soit appliqué, il faut que :

1. le `topic` courant soit filtré par `*` ;
2. l'entrée reçue soit filtrée par `* CARTE BLEUE` ;
3. et la sortie précédente soit filtrée par `_`.

L'un des aspects les plus intéressants en AIML est la balise `srai`. Elle permet d'intégrer la récursivité dans le langage en réinjectant une entrée dans l'algorithme de filtrage AIML. On peut ainsi à partir d'un `template` appeler séquentiellement d'autres `template`. Il devient alors possible de réaliser des instructions conditionnelles et des séquences d'instructions.

Ce principe est illustré par la Fig. 4. Un appel à cette `category` fixe la valeur du `topic` et du `produit`. Les traitements correspondant au `pattern` `INITIALISATION` (cf. Fig. 5) et au `pattern` `TROUVER` (cf. Fig. 6) sont ensuite effectués séquentiellement.

```

1 <category>
2   <pattern>_ VELO</pattern>
3   <template>
4     <set name="topic">trouver velo
5     </set>
6     <set name="produit">>null</set>
7     <srai>INITIALISATION</srai>
8     <srai>TROUVER</srai>
9   </template>
10 </category>

```

FIGURE 4 – Exemple d'utilisation de `srai`

## 4.2 Étude de faisabilité

Dans cette section, nous testons AIML afin de savoir s'il permet de mettre en œuvre les caractéristiques d'un PDA identifiées dans la section 3. AIML est destiné à la conception d'agents conversationnels réactifs. Ainsi, le système de `pattern-template` convient naturellement pour implémenter la réactivité de notre PDA. Cependant, ses aspects proactifs s'appuient davantage sur le contexte dialogique. Leur implémentation n'est donc pas évidente en AIML. Nous évaluons ici la pertinence de ce langage dans ce cadre.

L'initiative est implémentée via un module externe qui, en fonction de ses règles de déclenchement, invoque un `template` d'initiative par l'intermédiaire d'un `pattern` prévu à cet effet. Ainsi, le `timeout` est une entrée spéciale, qui active un `template` associé.

Nous assimilons un but à un contexte dialogique particulier. Le mécanisme d'adoption de buts par notre PDA est implémenté à l'aide de l'élément `topic` en AIML. Quand le PDA adopte un but, il change de contexte et donc de `topic`. Le `template` de la règle dialogique dans la figure 4 fixe ainsi le `topic` à `trouver velo`. Les règles de dialogues incluses dans tout `topic` filtrant le `topic` courant sont alors activées.

Un PDA doit orienter le dialogue afin d'acquiescer l'information qui lui est utile. Cependant, poser des questions à l'utilisateur n'est pas naturel avec le système de `pattern-template` d'AIML. La difficulté réside notamment dans le choix des questions. En effet, celles-ci doivent porter sur des informations qui ne sont pas déjà connues. A cette intention, nous utilisons la mémoire du `bot` qui est constituée d'attributs. Il faut tout d'abord effectuer une instanciation et une initialisation des attributs à la valeur nulle (cf. Fig. 5).

```

1 <topic name="trouver velo">
2   <category>
3     <pattern>INITIALISATION</pattern>
4     <template>
5       <set name="sexe">null</set>
6       <set name="age">null</set>
7       <set name="tailleVelo">null
8     </set>
9       <set name="budget">null</set>
10      <set name="couleur">null</set>
11    </template>
12  </category>
13 </topic>

```

FIGURE 5 – Catégorie d’initialisation

Une fois l’initialisation réalisée, l’algorithme principal est exécuté (cf. Fig 6). Le comportement du PDA se décompose en trois étapes. Primo, le PDA essaie d’inférer des informations à partir des nouvelles données éventuellement acquises (DEDUIRE). Secundo, il suggère si possible un produit (PROPOSER). Tertio, il amorce la collecte de nouvelles données (ACQUERIR).

```

1 <category>
2   <pattern>TROUVER</pattern>
3   <template>
4     <condition>
5       <li name="produit"
6         value="null">
7         <srai>DEDUIRE</srai>
8         <srai>PROPOSER</srai>
9         <srai>ACQUERIR</srai>
10      </li>
11     <li>
12       Dans le panier :
13       <get name="produit"/>.
14     </li>
15   </condition>
16 </template>
17 </category>

```

FIGURE 6 – Catégorie principale

À chaque question correspond un attribut. Il est ensuite possible de tester leur valeur avec la balise `condition`. Si celle-ci est nulle, on pose alors la question associée. Le bot acquiert ensuite l’information contenue dans l’entrée et affecte la valeur correspondante à l’attribut.

Nous dotons le PDA de la capacité d’adaptabilité à l’aide de l’élément AIML `topic`. Dans le cas où l’utilisateur ne sait pas répondre à une question, le `topic` est modifié afin d’activer les règles dialogiques pour l’affectation de l’attribut. Une fois l’attribut déduit des nouvelles informations collectées, le `topic` précédent est

```

1 <topic name="taille velo">
2   <category>
3     <pattern>DEDUIRE</pattern>
4     <template>
5       <condition name="hauteur">
6         <li value="1m05">
7           <set name="tailleVelo">14
8         </set>
9         <set name="topic">trouver
10        velo
11      </set>
12    </li>
13    <li value="1m10">
14      <set name="tailleVelo">16
15    </set>
16    <set name="topic">trouver
17    velo
18  </set>
19  </li>
20 </condition>
21 </template>
22 </category>
23 </topic>

```

FIGURE 7 – Implémentation de règles en AIML

réactivé (cf. Fig. 7).

La mise en place d’énoncés par défaut pour assurer la robustesse de l’agent est implémentable en AIML. Il est possible de produire une `category` d’échappement pour chaque `topic` à l’aide du motif `*` qui possède la priorité la plus faible. On peut alors selon les cas effectuer une demande de reformulation auprès de l’utilisateur ou appeler récursivement un `template` en particulier pour poursuivre la conversation.

Comme présenté dans la figure 3, la flexibilité est implémentée en utilisant la balise `that` et le motif «`_`» de manière à garder une priorité sur les autres catégories. Une fois le traitement effectué, la balise AIML `srai` permet de rappeler récursivement l’algorithme de filtrage afin de réactiver le contexte précédent (`trouver velo` lignes 9 et 15 dans la Fig. 7).

### 4.3 Limites

Notre implémentation d’un PDA reste limitée de par l’usage du langage AIML.

Le langage d’expressions régulières pour définir les motifs de filtrage en AIML est primitif. Il n’existe par exemple pas de motif pour filtrer une valeur comprise dans un ensemble donné, e.g. un entier. Le joker permet de filtrer l’ensemble de toutes les suites de mots mais sans distinction. Ainsi, le motif `Je cherche un`

vélo \* filtre indistinctement les énoncés « Je cherche un vélo *rouge* » ; « Je cherche un vélo *14"* » ; « Je cherche un vélo *pas cher* ».

Il n'est donc pas possible de proposer un traitement différent selon la nature de l'information exprimée. Ceci rend délicate l'acquisition de données.

Plus encore, c'est la fiabilité même de l'information collectée qui est remise en question. En effet, si on pose par exemple une question visant à acquérir un âge, on adoptera un motif de la forme \* ans pour extraire le nombre d'années de la réponse de l'utilisateur. Mais la formulation de sa réponse ne correspondra peut-être pas et on prend alors le risque de sauvegarder une donnée erronée en mémoire.

Par exemple :

5 ans → \* = « 5 »

J'ai 5 ans → \* = « J'ai 5 »

Ici, dans le second cas, la chaîne attribuée à l'attribut age sera « J'ai 5 ». Pour réduire cet effet, on peut multiplier les motifs (et donc les catégories) pour envisager un maximum de possibilités. Cependant, on arrive très vite à un nombre de règles de dialogue important. Plus de 30 règles ont ainsi été produites pour notre prototype se bornant à couvrir la conversation en figure 1. L'exploration de ces règles devient vite difficile et on perd en extensibilité.

AIML manipule uniquement des chaînes de caractères, il n'est donc pas possible de contrôler si l'information capturée est bien du type attendu. Les seuls tests possibles sont des tests d'égalité entre deux chaînes. La comparaison de valeurs numériques n'est pas réalisable. De plus, AIML ne permet pas de manipuler des structures de données (enregistrement, tableaux, listes, ...). Il n'est donc pas possible de lier explicitement des attributs à un produit, dans notre exemple la couleur, la taille ou le prix au vélo.

Les limitations fortes présentées dans cette section nous convainquent de l'inadéquation d'AIML avec nos besoins pour mettre en œuvre notre PDA.

## 5 Travaux connexes

De nombreux travaux traitent des agents conversationnels. Une des approches consiste à les enrichir de capacités multimodales. Ainsi, l'outil DIVA [8] permet de munir les agents de capacités déictiques afin de guider l'utilisateur dans

l'interface sur une page Web. Les agents sont alors directement intégrés à la structure de la page. Riviere et al. [16] proposent une librairie d'actes de langage adaptés à l'expression multimodale d'un ACA. S'appuyant sur la théorie des actes de discours de Searle et Vanderveken, ces travaux font le lien entre des émotions dites complexes et les actes de langages au travers des conditions de sincérité, du mode d'accomplissement et du degré de puissance. Les actes de langages proposés ont été testés à l'aide de l'ACA appelé GRETA développé par l'équipe de Catherine Pelachaud [15]. Les travaux de Barthès [7] se focalisent le problème de l'interprétation du langage naturel pour l'élaboration d'un agent assistant. Ayant circonscrit un ensemble de tâches, [7] définit une ontologie du domaine et un ensemble de gabarits (en anglais *pattern*) qui permettent d'identifier la tâche demandée à l'agent.

Dans le champ des ACAs, le terme proactif a pu être employé avec différentes nuances. Pour L'Abbate et al., la proactivité est un état de l'agent [11]. Un *chatbot* peut ainsi au moment opportun, c'est-à-dire quand il dispose de l'information nécessaire, passer d'un état réactif à un état proactif. Une fois dans l'état proactif, il fera usage de l'information dont il dispose pour mieux adapter son comportement à la situation. Pour Semaro et al. [17], un agent proactif est capable de poursuivre des buts dans une conversation, par exemple en recommandant des produits à un client en particulier. A la différence de notre approche, il utilise dans ce cas des informations sur le client stockées au préalable pour lui proposer des produits qui lui correspondent.

Il existe d'autres outils pour le déploiement d'agents conversationnels, comme NabuTalk et CSO-LP. CSO-Language Processor d'Artificial Solution [4] est une technologie reposant sur des règles d'interaction. Ces règles se basent sur les entrées et l'état du dialogue pour définir une sortie appropriée. NabuTalk de Dialonics [5] inclut du traitement sémantique et permet la construction de motifs évolués. En ce sens, il répond pour bonne partie aux limites d'AIML. Ces deux outils sont cependant propriétaires, ce qui ne satisfait pas nos contraintes.

## 6 Discussion

Le dialogue entre un agent humain et un agent artificiel permet l'échange d'information dans le but de répondre aux besoins de l'utilisateur. Dans une approche classique, l'agent se

contente de délivrer de l'information uniquement en réaction aux questions de l'utilisateur. Nous proposons ici un agent dialogique muni d'un comportement proactif lui permettant de jouer le rôle de conseiller. Il mène la conversation afin d'identifier les besoins de l'utilisateur et d'y répondre. En partant d'un exemple concret, nous avons identifié les principales caractéristiques que doit exhiber un PDA : initiative, réactivité, capacités à adopter des buts, orienter le dialogue et acquérir des informations, adaptabilité, robustesse et flexibilité.

Notre tentative de mise en œuvre en AIML d'un agent présentant ces caractéristiques nous a convaincus de l'insuffisance de ce langage pour cet objectif. En particulier, la faible expressivité des motifs et l'impossibilité de typer des informations dans un filtre constituent un obstacle fort. La définition d'un langage adapté, éventuellement en tant qu'extension d'AIML, s'impose donc.

Dans de futurs travaux, nous envisageons de munir le PDA d'une caractéristique supplémentaire, la persuasion. L'utilisation d'argument, comme la garantie d'un produit (cf. énoncé 18) peut convaincre le client pour conclure la vente. En plus de prendre en compte les buts de l'utilisateur, le PDA pourrait dans le même temps poursuivre d'autres buts, comme ceux de l'enseigne.

## Remerciements

Ce travail est soutenu par le projet Vendeur Virtuel Ubiquitaire (VVU) labellisé par le Pôle de Compétitivité des Industries du Commerce.

## Références

- [1] L'agent conversationnel Laura sur le site bleu ciel d'EDF. <http://bleuciel.edf.com> Dernière consultation le 26 avril 2011.
- [2] Site d'AIML. <http://www.alicebot.org/aiml.html> Dernière consultation le 26 avril 2011.
- [3] Site d'ALICE. <http://alicebot.blogspot.com/> Dernière consultation le 24 juin 2011.
- [4] Site d'Artificial Solutions. <http://www.artificial-solutions.com> Dernière consultation le 26 avril 2011.
- [5] Site de Dialonics. <http://www.dialonics.com> Dernière consultation le 26 avril 2011.
- [6] Site du prix Loebner. <http://www.loebner.net> Dernière consultation le 26 avril 2011.
- [7] Barthès J-P. A. Traitement de dialogues en langage libre par un agent assistant et son staff. In Valérie Camps and Philippe Mathieu, editors, *JFSMA'07 - Systèmes Multi-Agents - Modèles de comportements*, pages 191–200. Cépaduès, 2007.
- [8] Braffort A., Sansonnet J. P., Martin J. C., and Verrecchia C. DIVA, une architecture pour le support des agents gestuels interactifs sur internet. *TSI*, Vol. 29 (7) :777–806, 2010.
- [9] Beaufils B., editor. *État de l'art Informatique*. Livrable Projet Vendeur Virtuel Ubiquitaire, 2010.
- [10] Weizenbaum J. ELIZA - A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, Vol. 9 :36–45, 1966.
- [11] Marcello L'Abbate, Ulrich Thiel, and Thomas Kamps. Can proactive behavior turn chatterbots into conversational agents?. In *IAT'05*, pages 173–179, 2005.
- [12] James C. Lester, Sharolyn A. Converse, Susan E. Kahler, S. Todd Barlow, Brian A. Stone, and Ravinder S. Bhogal. The persona effect : affective impact of animated pedagogical agents. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 359–366. ACM, 1997.
- [13] Maxime Morge, Sameh Abdel-Naby, and Bruno Beaufils. Towards a dialectical approach for conversational agents in selling situations. In *Proc of ArgMAS 2010*, number 6614 in LNAI, pages 141–158. Springer, 2011.
- [14] Ben Mimoun M.S. and Poncin I. Agents virtuels vendeurs : que veulent les consommateurs? In *WACA'10 : 4ème Workshop sur les Agents Conversationnels Animés*, pages 99–106, 2010.
- [15] I. Poggi, C. Pelachaud, F. Rosis, V. Carofiglio, and B. Carolis. Greta. a believable embodied conversational agent. In Oliviero Stock and Massimo Zancanaro, editors, *Multimodal Intelligent Information Presentation*, volume 27 of *Text, Speech and Language Technology*, pages 3–25. Springer Netherlands, 2005.

- [16] J. Rivière and S. Pesty. Actes de langage et emotions : vers un langage de conversation multimodal. In *JFSMA'10 : Journées Francophones des Systèmes Multi-Agents*, pages 107 – 116, 2010.
- [17] Giovanni Semeraro, Hans H. K. Andersen, Verner Andersen, Pasquale Lops, and Fabio Abbattista. Evaluation and validation of a conversational agent embodied in a bookstore. In *Proceedings of the User interfaces for all 7th international conference on Universal access : theoretical perspectives, practice, and experience, ERCIM'02*, pages 360–371, Berlin, Heidelberg, 2003. Springer-Verlag.
- [18] D. Walton and E. Krabbe. *Commitment in Dialogue*. SUNY Press, 1995.