

# Deux plates-formes pour l'expérimentation d'un agent conversationnel animé

*Yannick Fouquet, Jean Caelen*

Laboratoire CLIPS-IMAG, B.P. 53, 38041 Grenoble cedex 9, France

[Yannick.Fouquet@imag.fr](mailto:Yannick.Fouquet@imag.fr), [Jean.Caelen@imag.fr](mailto:Jean.Caelen@imag.fr)

## ABSTRACT

This paper presents 2 platforms for testing man-machine spoken dialog system in which an embodied conversational agent is in charge of interaction management. The agent used is expressive and gestural and as been applied to e-commerce. The first platform is intended to test each sequence of the interaction, each movement, each expression, each sentence. The second one uses a wizard of Oz which manipulates the agent, its expressions, gestures, statements, etc. The user thinks dialoging with a computer. It let us collect corpus in situation and test usability and acceptability of the agent.

## RÉSUMÉ

L'article présente nos travaux sur deux plates-formes d'expérimentation de dialogue homme-machine (éventuellement oral), avec pour exemple un agent conversationnel expressif et gestuel dans le domaine du commerce électronique. La première concerne les tests unitaires, c'est-à-dire l'évaluation de chaque séquence (et même chaque geste, expression et énoncé) de dialogue avec un agent conversationnel animé. La deuxième fait intervenir un magicien d'Oz pour manipuler l'agent (gestes, expressions, énoncés, etc.) à l'insu de l'utilisateur et recueillir ainsi du corpus en situation.

## 1. INTRODUCTION

Cet article présente la mise en place de deux plates-formes d'expérimentation pour l'évaluation de dialogues avec un agent conversationnel animé. La deuxième considère le magicien d'Oz comme une solution acceptable pour participer à l'évaluation de dialogues multimodaux homme-machine.

Un agent conversationnel peut remplir différentes fonctions selon les objectifs qui lui sont assignés. Il peut animer une communication interne, assister, conseiller, former, etc. L'objectif scientifique du projet ACE (Agent Conversationnel Expressif) est d'intégrer un ensemble d'éléments de communication non verbale à un dialogue multimodal dans le but d'améliorer l'efficacité de l'interaction entre un agent conversationnel et ses utilisateurs. Son originalité réside dans le développement d'un modèle de communication gestuelle générique, *piloté* par le dialogue et *synchronisé* avec les étapes de celui-ci, qu'il s'agisse de dialogues dirigés par l'agent conversationnel ou dirigés par l'utilisateur. Dans le premier cas, l'agent montre une continuité expressive au fur et à mesure de la progression du scénario de dialogue, dans le second cas, il exprime des émotions en réaction aux répliques de l'utilisateur.

Le projet a comporté trois étapes importantes. Nous présenterons les **tests unitaires** élaborés pour évaluer l'agent conversationnel animé existant. Ceux-ci portent en particulier sur sa compréhensibilité, sa compétence perçue et les attentes de l'utilisateur face à un tel système de dialogue. Une simulation avec une comédienne a ensuite été réalisée pour capturer les paramètres comportementaux de l'agent en situation. L'analyse du film obtenu a permis l'élaboration d'un nouvel agent communicationnel expressif. Pour l'**évaluation ergonomique** de ce nouvel agent, nous avons développé une plate-forme expérimentale de type magicien d'Oz intégrant la communication verbale et non verbale. Dans celle-ci, un compère manipule l'agent conversationnel, faisant passer ce dernier pour un avatar aux yeux des sujets. Les résultats sont montrés dans [Golanski & Caelen 2005].

## 2. UNE PLATEFORME POUR DES TESTS UNITAIRES, UN SYSTÈME NON INTERACTIF

Les premiers tests concernent chaque élément de l'agent pris indépendamment des autres. Nous cherchons à évaluer les éléments du dialogue, ceux concernant les expressions de l'agent, et ceux concernant ses gestes. Ces tests ont la forme d'un questionnaire (figure 1) qui se déroule ainsi :

- ✓ Une séquence contenant du dialogue et un avatar expressif gesticulant est présentée.
- ✓ Des questions sont ensuite posées sur :
  - la compréhension de la séquence (test DQR, Dialogue-Question-Réponse)
  - les attendus ; ce que le sujet pense qu'il va se passer ensuite (test DAR, Dialogue-Attendu-Reponse)
  - les attentes ; ce que le sujet aimerait qu'il se passe ensuite (tests UARD, Utilisateur-Attente-Response-Dialogue).



The screenshot shows a web browser window with the address bar containing `http://euripide.imag.fr/ace/index.php`. The page title is "Angela - tests unitaires". The main content area displays a 3D avatar of a woman with short black hair and a black top. To the right of the avatar, the text reads: "Bonjour, ici l'assistante virtuelle Vocalisa. Quelle est votre demande ?". Below this, there is a list of five questions, each followed by a checkbox and the word "oui":

L'agent a-t'il salué avec la tête ?	<input type="checkbox"/>	oui
L'agent a-t'il fait une suggestion ?	<input type="checkbox"/>	oui
L'agent a-t'il fait une proposition ?	<input type="checkbox"/>	oui
L'agent a-t'il émis un avertissement ?	<input type="checkbox"/>	oui
L'agent est-il amical ?	<input checked="" type="checkbox"/>	oui

At the bottom of the form is a button labeled "valider". The browser's status bar at the bottom shows the XML declaration: `<?xml version="1.0"`.

**Figure 1** : Un système non interactif de tests unitaires

Ces tests, développés en php et xml, peuvent être mis en place sur un serveur quelconque, et être réalisés à partir d'un ordinateur quelconque connecté à internet. Les sujets peuvent ainsi être très divers. La seule contrainte est de pouvoir afficher l'agent « Angela » (windows pour le moment).

L'utilisateur se connecte au système *via* une page web (php) contenant du code (javascript) permettant de manipuler localement l'agent conversationnel. L'agent animé se charge et lui apparaît. Il peut alors débiter la série de tests unitaires.

Chaque test correspond à une page (figure 1). Lorsqu'une page s'affiche, une séquence (notée D pour Dialogue) est jouée, mettant en jeu l'agent conversationnel, son énoncé, ses expressions et ses gestes. Ces séquences D proposées sont en format XML, incluant une annotation en actes et états affectifs. Ce pourra être du NVML ou du APMML [Pelachaud, 03] par exemple :

```
<APML ouverture type>
  <turn-allocation type = prise de parole>
    < performative type = salutation>
      Bonjour, je suis Angela
    </performative>
  </ turn-allocation>
  <affective type = enjoué>
    C'est <topic-comment type = phatique> formidable </topic-comment> de vous revoir
  </affective>
  <certainty type= certitude>
    J'étais sûre que vous alliez rappeler
  </certainty>
</APML ouverture>
```

S'ensuit une série de questions posées sous forme de formulaire. Ces questions portent sur des éléments de la séquence annotée jouée. Elles permettent de définir la compréhensibilité de l'agent, sa compétence perçue et les attentes que l'on peut avoir d'un tel système. Elles sont donc de trois ordres : compréhension, attendus, attentes. Après avoir répondu, il valide et passe à la séquence suivante.

### 2.1 Compréhension, le paradigme DQR (Dialogue-Question-Réponse)

Les tests de compréhension de la séquence (DQR) peuvent porter sur l'acte ou sur l'état 'psychologique' de l'agent. Ils auront la forme xml suivante :

```
<dqr id="suggest">
  <question reponse="oui">L'agent a-t'il fait une suggestion ? </question>
  <question reponse="non">L'agent a-t'il fait une proposition ? </question>
  <question reponse="non">L'agent a-t'il émis un avertissement ? </question>
</dqr>
<dqr id="joy">
  <question reponse="oui">L'agent est-il joyeux ? </question>
  <question reponse="non">L'agent est-il triste ? </question>
  <question reponse="non">L'agent est-il stressé ? </question>
</dqr>
```

On présente alors un dialogue **D** à un utilisateur, puis une question **Q** dont on connaît la réponse « idéale » **R**. On la compare avec celle de l'utilisateur **R<sub>u</sub>**.

D = <Séquence d'ouverture avec un agent joyeux>	ou	D = <Séquence d'ouverture avec un agent joyeux>
Q = est-ce un dialogue d'ouverture ?		Q = l'agent est-il enjoué ?
R = oui		R = oui
	mais	D = <Séquence d'ouverture avec un agent joyeux>
		Q = l'agent est-il triste ?
		R = non

On s'intéresse donc à ce qu'il a perçu de la séquence. A-t-il perçu le geste, l'expression, l'acte, etc. ? Cela permet de repérer les éléments utiles à l'interaction. Par exemple, un agent, qui montre avec ses yeux alors que l'utilisateur ne perçoit pas les yeux, devra davantage se tourner vers un geste déictique plus visible (tourner la tête ou montrer avec le doigt par exemple).

## 2.2 Attendus, le paradigme DAR (Dialogue-Attendu-Réponse)

Le test sur les attendus fonctionne comme le test DQR, on demande seulement à l'utilisateur ce qui va se passer après la séquence dialogique jouée et on compare sa réponse avec l'action effectuée par l'agent.

```
<dar id="1" dialogue="corpus_test_unitaires.xml">
  <attendu reponse="non">Est-ce un préambule à une aide ? </attendu>
</dar>
```

Nous pourrions ainsi proposer des questions de la forme :

D = <Séquence d'ouverture avec un agent joyeux>  
A = est-ce un préambule à une aide ?  
R<sub>U</sub> = oui (R=non, car l'agent est programmé pour demander le profil utilisateur)

Ce test reflète donc ce que pense l'utilisateur que la machine peut faire. On peut ainsi repérer la compétence perçue du système afin de l'améliorer. Par exemple, si aucun énoncé n'est préambule à une aide alors que l'agent possède une séquence d'aide alors il faut créer un tel énoncé.

## 2.3 Attentes, le paradigme UARD (Utilisateur-Attentes-Réponse-Dialogue)

Enfin les tests sur les attentes ([Fouquet 2004b]) reflètent ce que l'utilisateur voudrait que la machine fasse. L'utilisateur a l'initiative, il produit un énoncé **U**, on lui pose une question **A** sur ses attentes et on compare sa réponse **R<sub>U</sub>** avec l'action **D** de l'agent. Ces questions peuvent être posées sur l'acte attendu, sur l'état 'psychologique' (ou expression) attendu ou sur le geste attendu :

```
<uard id="1" dialogue="corpus_test_unitaires.xml">
  <utilisateur> Bonjour je suis Yannick Fouquet </utilisateur>
  <attente reponse="non"> Vous attendez vous à une présentation de l'agent ? </attente>
</uard>
```

Nous pourrions ainsi proposer des questions de la forme :

U	« bonjour, je suis Yannick Fouquet »		
A	vous attendez vous à une présentation de l'agent ?	vous attendez-vous à une ouverture sérieuse de l'agent ?	vous attendez-vous à un mouvement des bras de l'agent ?
R <sub>U</sub>	oui	oui	oui
D	<Séquence d'ouverture avec un agent joyeux>		
	R=oui (salutation)	R=non (enjoué)	R=non (pas de geste)

Ces derniers tests permettent de répondre, pour chaque séquence, à la question : « le système répond-t-il aux attentes de l'utilisateur ? » (R<sub>U</sub>=R ?) et reflètent les souhaits des utilisateurs.

Ces trois types de tests permettent ainsi l'évaluation de chaque portion du modèle de dialogue afin d'améliorer la pertinence et l'efficacité des aspects verbaux et non-verbaux de l'agent conversationnel animé. Les séquences D incluent une annotation en actes et états affectifs. Les questions posées sont alors adaptées automatiquement aux informations recueillies dans les séquences (ex: <affective type = enjoué> donc questions sur l'état affectif sérieux, enjoué, etc.) afin de déterminer les faiblesses du système et de les résoudre.

### 3. UNE PLATEFORME POUR DES TESTS ERGONOMIQUES, UNE SIMULATION DU SYSTÈME

La plate-forme d'expérimentation de dialogues, avec un agent conversationnel animé par un compère, est séparée en trois applications : l'interaction avec l'utilisateur, la copie clone de cette interaction, et la simulation du système proprement dit. Chacune peut être exécutée sur n'importe quel ordinateur du réseau. Le langage de description utilisé pour manipuler l'agent conversationnel, *via* les scénarios, est le NVML mis en place pour le projet ACE. Pour la manipulation élémentaire, c'est un format texte reprenant le type d'action (**E**xpression, **A**nimation, **P**arole, etc.) suivi de l'action correspondante. La plate-forme, intermédiaire entre le système (ici simulé) et l'agent animé, accepte n'importe quel format de description (texte ou xml) importé avec les scénarios prédéfinis.

#### 3.1 Interaction avec l'utilisateur, un client.

Cette première application concerne l'utilisateur et son interaction avec le système. C'est une page web contenant du code (javascript) permettant de manipuler localement l'agent conversationnel (développé par la Cantoche) et une applet (java) permettant de gérer le dialogue avec le système. Cette applet est connectée au système par le biais d'une socket (connexion point à point). L'utilisateur peut alors interagir par l'oral (à condition de posséder un système de reconnaissance vocale comme ELAN), mais aussi par écrit en saisissant son texte dans une zone prévue à cet effet dans la fenêtre de dialogue. L'historique des énoncés du dialogue est également affiché.

L'utilisateur dispose de 3 zones d'interaction. Il peut parcourir les pages web du site par le biais d'une fenêtre internet. Une deuxième fenêtre s'affiche automatique en avant du site avec le minimum de décoration possible. Elle permet l'interaction avec le système par le biais de l'écrit. Cette fenêtre possède une zone de saisie et une zone affichant tous les messages répondus par le système de sorte que l'utilisateur peut consulter l'historique de la conversation. La troisième zone d'interaction est l'agent conversationnel animé lui-même. Il permet de représenter le système sous forme d'un avatar expressif et gesticulant et ainsi enrichir la communication par des aspects non verbaux (gestes déictiques, icôniques, métaphoriques ou phatiques, expressions, animations, etc.). Il peut accepter l'interaction vocale par le biais de systèmes de reconnaissance et de synthèse de parole. En voici un exemple en figure 2 :

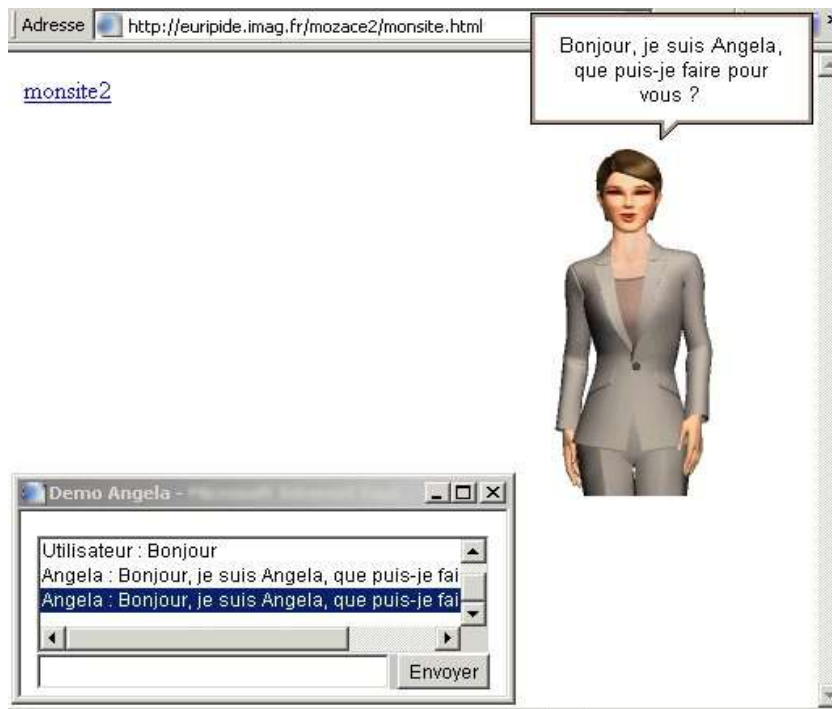
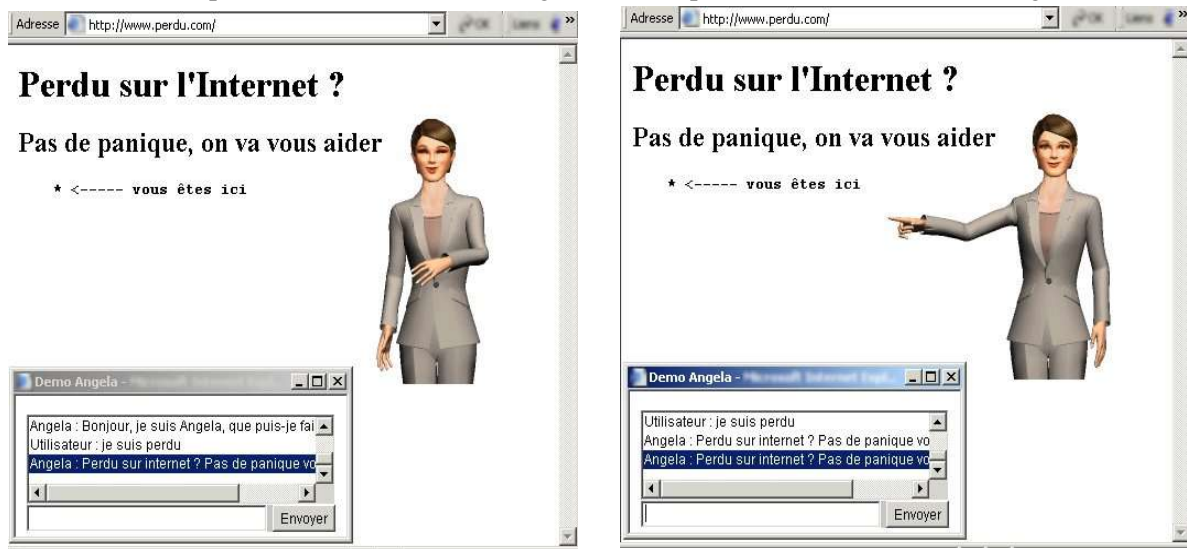


Figure 2 : Un client interactif pour l'utilisateur

Séparer ces 3 zones permet à l'utilisateur de les déplacer et les positionner comme bon lui semble sur l'écran. Nous pouvons ainsi déterminer les positions à préférer par défaut. L'agent doit alors connaître sa position afin d'effectuer les gestes déictiques corrects comme sur la figure 3 :



**Figure 3** : Une séquence contenant gestes, expressions et énoncé

Lorsqu'une séquence de réponse est envoyée par le système, le client la décompose en actions élémentaires. Chaque action est alors effectuée. L'énoncé est affiché dans l'historique et le client manipule l'agent conversationnel. Celui-ci prononce l'énoncé dans le cas d'une interaction orale, affiche l'énoncé dans une bulle, ouvre la page web, adopte l'expression et joue l'animation gestuelle inscrits dans la séquence. Ainsi mis en place, le client est indépendant du système lui-même auquel il est juste lié par une socket et un protocole d'interaction. Le système peut alors être réel ou simulé.

### 3.2 Copie clone de l'interaction, un client.

Le système est ici simulé par magicien d'Oz. Le compère utilise deux applications indépendantes pouvant être affichées sur deux écrans voire deux ordinateurs différents. L'une est la même que celle de l'utilisateur et permet au compère de voir ce que l'utilisateur voit. Cette solution a été préférée à un système de prise de contrôle à distance nécessitant de l'utilisateur qu'il soit situé dans un réseau local ou du moins qu'il n'y aie aucune barrière (pare-feu) entre celui-ci et le système.

Ce clone possède exactement la même interface que le client (en fait, c'est la même application). Il est identifié comme clone par son adresse IP connue du système à l'avance (par exemple la même adresse que le serveur) contrairement au client. Ainsi, le système se contente de dupliquer vers ce clone tout ce qu'il reçoit du client et ce qu'il lui répond. En outre, il n'écoute pas ce que ce clone pourrait vouloir lui envoyer de sorte que ce clone fonctionne uniquement comme interface de sortie pour le système.

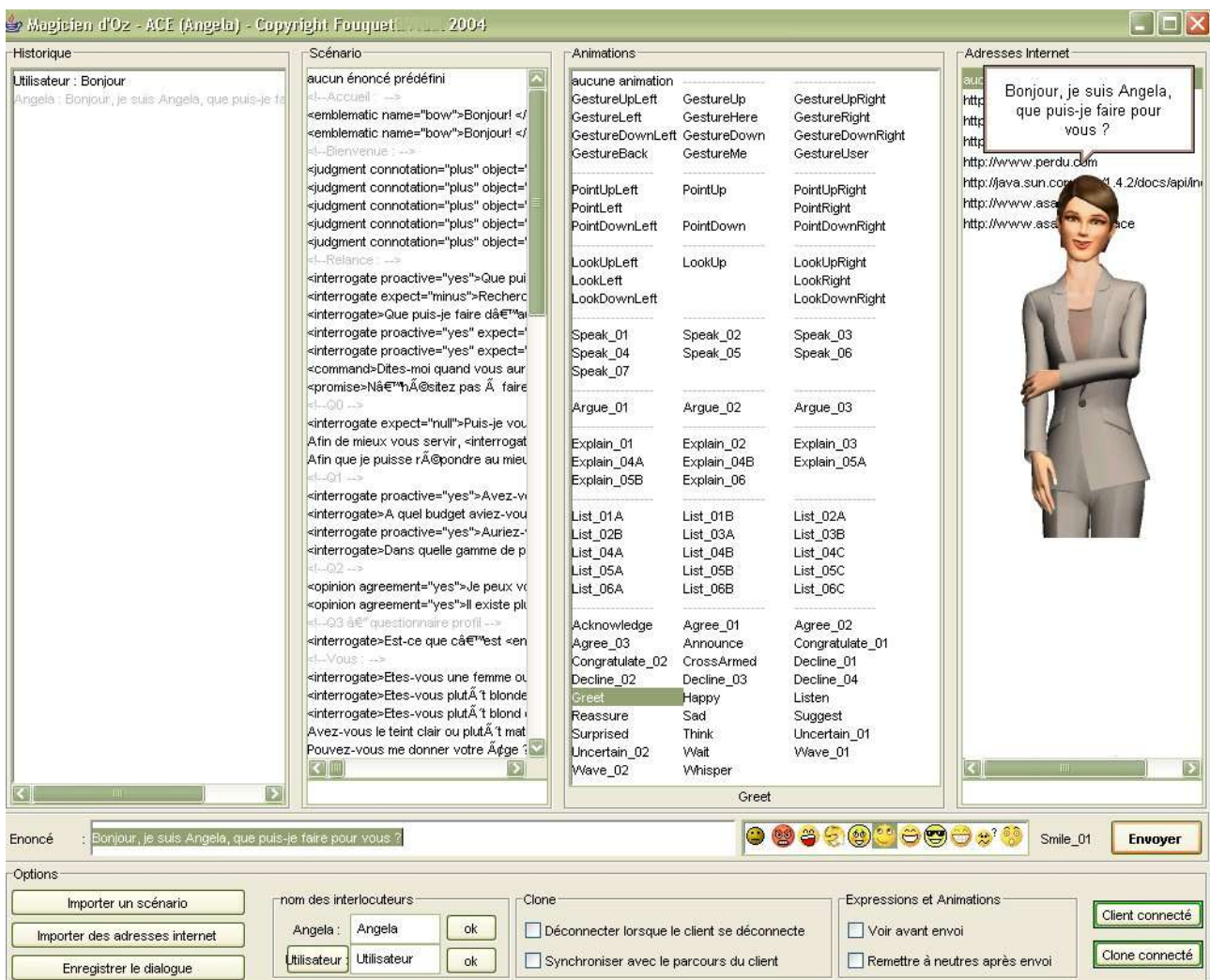
### 3.3 Système, un serveur.

La dernière application est le système lui-même. A l'instar de [Fouquet 2003] et [Fouquet 2004a] pour l'oral, c'est un serveur (développé ici en java) sur lequel se connectent l'utilisateur et le clone. Son interface est suffisamment générique pour pouvoir évaluer différents types de dialogues. En effet, l'expérimentateur y importe les éléments spécifiques : adresses de sites internet que l'agent pourra afficher à l'utilisateur, et scénarios de dialogues prédéfinis. Le système est donc indépendant du site sur lequel l'agent est évalué. Cette importation permet de produire des énoncés génériques ou applicables à grand nombre d'applications de commerce électronique par exemple, et des énoncés propres à l'application. Par exemple, un site de conseils beauté aura besoin d'informations sur le sexe ou le type de peau de l'utilisateur alors qu'un site de réservation en ligne n'en aura *a priori* que faire.



Une fois l'interface mise en place (avec les énoncés prédéfinis et les adresses internet proposées importés) et le clone connecté, le système attend les utilisateurs. Lorsqu'un utilisateur se connecte, le dialogue peut commencer. Tant que le dialogue a lieu, nul autre utilisateur ne peut se connecter car le système est simulé par magicien d'Oz.

L'interface (figure 4) du système présente au magicien une zone concernant l'historique des énoncés des interactants, des gestes et des expressions de l'agent. Pour répondre à l'utilisateur, il possède une liste de réponses prédéfinies contenant des énoncés, des expressions et des animations. Il lui est également possible de saisir sa propre séquence de réponse. Il peut alors, de manière complémentaire mais non nécessaire, écrire son propre énoncé de réponse, sélectionner une animation gestuelle que devra jouer l'agent, choisir une expression qu'il devra adopter, et sélectionner (ou saisir) une adresse de site internet à envoyer au client afin que celui-ci l'affiche sur son navigateur. Lorsque la séquence de réponse est prête, il l'envoie à l'utilisateur et au clone afin que ceux-ci la jouent. Il attend ensuite l'énoncé de l'utilisateur mais peut à tout moment le relancer ou lui proposer quelque chose de manière proactive.



**Figure 4 :** un système par magicien d'Oz pour manipuler l'agent conversationnel

Il dispose en outre d'un certain nombre d'options afin de synchroniser la navigation du clone avec celle de l'utilisateur ou de visualiser la séquence avant son envoi afin de la vérifier et la modifier si nécessaire ou encore remettre Angela dans un état neutre après avoir joué sa séquence à l'utilisateur. En fin de dialogue, l'utilisateur se déconnecte et l'expérimentateur peut enregistrer l'historique de la conversation.

Un point commun entre ces trois applications est l'affichage possible d'une infobulle (comme sur la figure 5), en bas à droite de l'écran, à l'arrivée d'un énoncé. Ainsi, les interactants n'ont pas à guetter la réponse de leur interlocuteur. Cela permet à chacun de se replonger dans la session de dialogue en cours.



**Figure 5** : une infobulle pour alerter de l'arrivée de l'énoncé

Cette infobulle est particulièrement utile au magicien qui prépare déjà la réponse suivante et n'est pas nécessairement concentré sur l'historique de la conversation. Elle sert dans une moindre mesure à l'utilisateur qui peut être amené à effectuer d'autres actions en attendant la réponse du système. Il risque alors de ne pas voir les expressions et animations d'Angela mais est alerté que le système a répondu.

#### **4. CONCLUSION**

Nous avons présenté dans ce papier la mise en place de deux plates-formes d'expérimentation dont une contenant un module simulé par magicien d'Oz. Celles-ci ont été développées pour l'évaluation d'un agent conversationnel animé efficace, à la fois expressif et gesticulant. Elles ont été utilisées pour les tests d'acceptabilité et d'utilisabilité appliqués à l'agent conversationnel « Angela » (les résultats sont développés dans [Golanski & Caelen 2005]) lors du projet ACE. L'étape suivante consistera à développer le modèle de dialogue du système. Son efficacité d'interaction est très importante. Nous pensons donc y intégrer les aspects pragmatiques [Xuereb & Caelen 2004], multisession [Nguyen & Caelen 2004], et les attentes montrées dans [Fouquet 2004b].

#### **REMERCIEMENTS**

Le développement java du magicien d'Oz a été réalisé en partie avec Audrey Vidal.

#### **BIBLIOGRAPHIE**

- [Fouquet 2003] Fouquet Y. (2003), *Le magicien d'Oz pour du dialogue oral : expérience avec un assistant virtuel en entreprise*. Actes de Rencontre Jeunes chercheurs en Parole, Grenoble, septembre 2003.
- [Fouquet 2004a] Fouquet Y. (2004), *Prédiction d'actes et attentes en dialogue : expérience avec un assistant virtuel simulé*. Actes de TALN'2004, p. 193-198, Fès, avril 2004.
- [Fouquet 2004b] Fouquet Y. (2004), *Modélisation des attentes en dialogue oral*, thèse de doctorat d'informatique, université Grenoble I, octobre 2004.
- [Golanski & Caelen 2005] Golanski C., Caelen J. (2005), *Test d'utilisabilité de l'agent conversationnel « Angela »*, WACA, Grenoble, juin 2005.
- [Nguyen & Caelen 2004] Nguyen H., Caelen J. (2004), *Multi-session Management in Spoken Dialogue System*. C. Lemaître, C.A. Reyes, J.A. Gonzalez éditeurs, *Advances in Artificial Intelligence*, IBERAMIA 2004, Springer ed., 2004
- [Xuereb & Caelen 2004] Xuereb A., Caelen J. (2004), *Un modèle d'interprétation pragmatique en dialogue homme-machine basé sur la SDRT*, Actes de TALN'04, XIème Conférence sur le Traitement Automatique du Langage Naturel, ISBN 2-9518235-5-5, p. 505-514, Fès, avril 2004