

Des Interfaces Multimodales sur Internet

Luc Julia

STAR Laboratory - SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025, USA
julia@speech.sri.com

Adam Cheyer

Artificial Intelligence Center - SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025, USA
cheyer@ai.sri.com

RÉSUMÉ

Internet nous a fait peur. Pendant près de deux ans, nous avons bien cru que nous étions revenus au niveau zéro des interfaces homme-machine quand nous utilisions Internet. Aujourd'hui flexibilité et portabilité sont d'actualité grâce à Java qui supplante, petit à petit, les pauvres HTML et CGI. Nous proposons de décrire dans ce papier l'évolution que nous avons connue et les choix que nous avons fait afin de développer de vraies interfaces multimodales sur Internet en utilisant notre architecture à base d'agents.

MOTS CLÉS : Internet, Multimodalité, Agents.

INTRODUCTION

Si Internet donne en théorie accès à un nombre incalculable et impressionnant d'informations sur tous les sujets, le problème auquel se trouvent confrontés les utilisateurs est celui de la manière d'y accéder. Les interfaces disponibles depuis l'avènement du Web sont le plus souvent statiques, n'utilisant que des liens hypermedia (HTML), ou au mieux, des composants (menus, boutons, etc..) qui limitent l'interaction à son strict minimum (CGI). Depuis un peu plus d'un an, Java, un C++ facile à utiliser et générant un code pouvant tourner sur presque toutes les machines au travers d'un navigateur Internet, nous a apporté la flexibilité et la créativité qui faisait tant défaut depuis les débuts du Web. Au SRI International, nous avons décidé d'investir en étendant à Java le support de notre architecture à base d'agents. Cette décision est la première étape vers le portage d'applications complexes, intégrant des interfaces intelligentes, utilisant cette architecture.

ARCHITECTURE

L'Open Agent Architecture (OAA) est une architecture à base d'agents qui a entre autre pour points forts [1]:

- Ouverture : OAA supporte des agents écrits en diverses langages sur diverses plates-formes.
- Distribution : Les agents d'une même application peuvent tourner sur des machines différentes
- Extensibilité : Des agents peuvent être ajoutés à l'application pendant qu'elle tourne. Apportant ainsi dynamiquement à la communauté des agents de nouvelles capacités.

- Collaboration : L'interface étant implantée avec des agents, l'utilisateur lui-même est vu comme un agent du système. Ceci simplifie la création de systèmes ou plusieurs humains et agents automatiques coopèrent.
- Mobilité : Les applications OAA peuvent tourner sur des ordinateurs à faibles performances, seule l'interface sera locale, les agents plus lourds seront distants.

Cette dernière propriété est particulièrement importante dans la perspective d'applications à travers le réseau. Internet étant déjà surchargé, il est impensable de distribuer des programmes complexes entiers au travers de lignes à faibles débits.

Le développement d'OAA en Java a permis de porter très facilement sur le réseau les applications dont nous disposons. La différence essentielle à laquelle nous sommes alors confrontés est de définir l'ensemble des modalités disponibles sur le Web.

LES MODALITES

Historiquement, les applications que nous proposons sont multimodales, utilisant en particulier la parole et les gestes 2D [2] de façon synergique. Les media utilisés étant un microphone et un stylo.

Dans le contexte réseau dans lequel nous nous plaçons, véhiculer la parole est un vrai problème. Les API disponibles en Java ne permettent pas encore la gestion du son en entrée. Nous avons alors décidé, afin de garder la parole comme principale modalité [3], d'utiliser provisoirement le téléphone comme moyen de transport pour la voix.

En ce qui concerne la reconnaissance des gestes, le portage en Java des algorithmes de reconnaissance [4] a été extrêmement simple de part leurs natures modulaires. La possibilité de dessiner les gestes à la souris a été ajoutée afin de garder l'idée de portabilité de Java sur n'importe quel type de machine et en particulier sur des machines ne disposant pas de stylo ou autres écrans tactiles.

Une autre modalité, associée au stylo est l'écriture manuscrite. Afin de conserver cette possibilité dans ces programmes Java, nous avons intégré JOT qui est un

reconnaisseur, caractère par caractère, de type Graffiti, fourni par CIC [5].

Les modalités utilisables à travers le réseau étant les mêmes que celles normalement utilisées moyennant de faibles adaptations, nous sommes maintenant prêt à porter quelques unes de nos applications.

PREMIER ESSAI: ATIS

La première application utilisant la reconnaissance de la parole sur Internet a été installée sur le serveur public Web du SRI International il y a plus d'un an. Plus de 4000 utilisateurs ont déjà pu tester le reconnaisseur de la parole dans une tâche de réservation de voyage aérien (ATIS, figure 1) [6] grâce à un navigateur et un téléphone. Le reconnaisseur, sans apprentissage, acceptant la parole continue, a été spécialement étudié afin d'obtenir les meilleurs résultats possibles en dépit de la faible qualité acoustique des téléphones [7].

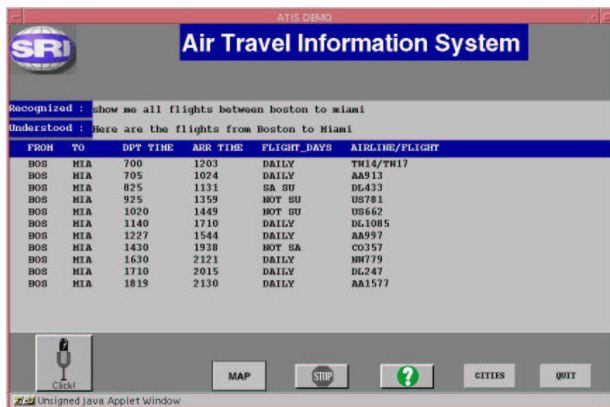


Figure 1: Interface Java pour ATIS.

A part quelques clics souris, cette première interface n'est en rien multimodale, mais elle a au moins l'avantage de tester la qualité du portage d'OAA et la faisabilité de l'intégration de la reconnaissance de la parole dans une interface Java pour le Web. Les résultats étant très satisfaisants, nous pouvons passer à l'étape suivante.

MULTIMODAL MAPS

Nous avons donc développé une interface multimodale en Java qui est une réplique exacte de l'interface tournant sur PDA [2]. Il s'agit en fait d'un nouvel agent qui vient s'insérer dans le système sans le perturber (figure 2). Les agents locaux, écrits totalement en Java, sont l'interface qui affiche les résultats et le reconnaisseur de gestes, ce sont les seuls éléments qui circuleront sur le réseau et qui tourneront sur la machine de l'utilisateur. Tous les autres agents, dont le reconnaisseur de la parole, restent sur le serveur du SRI.

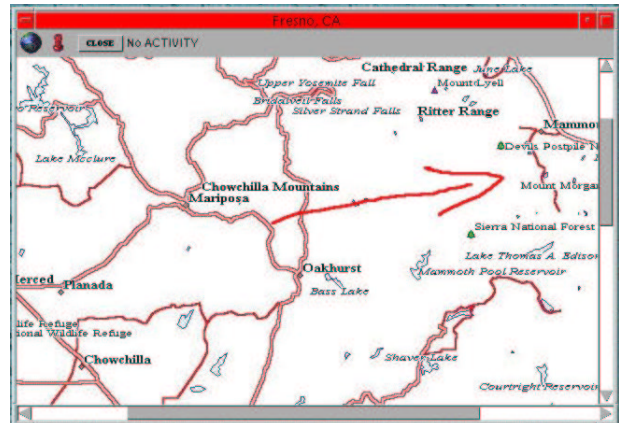


Figure 2 : L'interface de Multimodal Maps

Depuis nos efforts de portage en Java d'interfaces multimodales se multiplient. Certaines d'entre elles sont maintenant développées spécifiquement pour Java dès leur création. Ce sont ces interfaces que nous démontrerons lors de notre présentation.

BIBLIOGRAPHIE

1. Moran, Douglas B. and Cheyer, Adam J. and Julia, Luc E. and Martin, David L. and Park, Sangkyu. Multimodal User Interfaces in the Open Agent Architecture. In *Proceedings of UII'97* (Orlando, FL), 1997, pp. 61-68.
2. Julia, Luc and Cheyer, Adam. A Multimodal Computer-augmented Interface for Distributed Applications. In *Proceedings of HCII'95* (Tokyo, Japan), 1995, pp. 237-240.
3. Julia, Luc and Cheyer, Adam. Speech: A Privileged Modality. In *Proceedings of Eurospeech'97* (Rhodes, Greece), 1997, to appear.
4. Julia, Luc and Faure, Claudie. Pattern Recognition and Beautification for a Pen based Interface. In *Proceedings of ICDAR'95* (Montreal, Canada), 1995, pp 58-63.
5. <http://www.cic.com>
6. Julia, Luc and Cheyer, Adam and Neumeyer, Leo and Dowding, John and Charafeddine, Mehdi. [HTTP://WWW.SPEECH.SRI.COM/DEMOS/ATIS.HTML](http://WWW.SPEECH.SRI.COM/DEMOS/ATIS.HTML). In *Proceedings of AAAI'97 Spring Symposium* (Stanford, CA), 1997, pp. 72-76.
7. <http://www.nuancecom.com>