

X-CAMPUS : démarche et outils pour une assistance proactive contextuelle

Hajer Sassi¹, José Rouillard¹ et Jean-Claude Tarby¹

1 : Université de Lille 1, Laboratoire LIFL, 59655 Villeneuve d'Ascq Cedex, France.

Contact : Hajer.Sassi@ed.univ-lille1.fr

Résumé

La contribution des nouvelles technologies au développement de l'informatique a donné naissance à de nouveaux domaines de recherches tel que celui de l'Intelligence Ambiante (IAm), où les nouvelles technologies sont utilisées comme des outils pour assister les utilisateurs dans leurs tâches quotidiennes. Notre étude porte principalement sur l'assistance proactive et l'interaction homme-machine ; nous étudions l'assistance proactive contextuelle dont le but est de concevoir un système adaptatif capable de déterminer les besoins de l'utilisateur souhaitant disposer d'un assistant virtuel. Nous nous sommes intéressés à l'interaction proactive contextuelle, dans le cadre d'un dialogue entre un humain et un agent conversationnel en ligne nommé X-CAMPUS. X-CAMPUS est l'acronyme de *eXtensible Conversational Agent for Multichannel Proactive Ubiquitous Services*. Il a pour vocation d'assister l'utilisateur dans ses tâches quotidiennes grâce à sa capacité de percevoir l'état de l'environnement et d'interagir pertinemment selon les besoins de l'utilisateur. Nous montrons que selon le contexte, l'agent notifie l'utilisateur avec des messages personnalisés (ex : suggestion de restaurants selon leurs menus, en accord avec les préférences de l'utilisateur) via le canal le plus approprié (messagerie instantanée, e-mail et SMS).

Abstract

The contribution of new technologies in the development of the computing gave a rise to news domains of researches such as the Ambient Intelligent (Aml), where the new technologies are used as tools to assist users in their daily tasks. Our study mainly the proactive assistance and the interaction human-machine; we study the contextual proactive assistance in order to design an adaptive system able to determine the needs of user wishing have a virtual assistant. We are interested in the contextual proactive interaction between a human and a conversational agent, that we have named X-CAMPUS. X-CAMPUS stands for *eXtensible Conversational Agent for Multichannel Proactive Ubiquitous Services*. It aims to assist user in his daily tasks thanks to its ability to perceive the state of the environment and interact effectively according to the user's needs. We show that according to the context, X-CAMPUS notifies the user via personalized messages (e.g., suggestion of restaurants according to their menus and the users' preferences) and also according to the most appropriate channel (instant messaging, e-mail and SMS).

Mots-clés : Agent conversationnel, assistance proactive, intelligence ambiante, système adaptatif, multicanalité.

Keywords : Conversational agent, proactive assistance, ambient intelligent, adaptative system, multi-channel.

1. Introduction

Depuis dix ans, les évolutions technologiques, telles que les capacités de traitement et de communication de l'information se sont succédé pour jouer un rôle primordial dans le développement de l'informatique. Nous nous trouvons face à un monde numérique très riche en informa-

tion mais difficile à l'exploiter pour plusieurs raisons : confidentialité, sécurité, gestion des connaissances, prise de décisions de manière autonome, collaboration, etc. Par ailleurs, le domaine de l'intelligence ambiante (IAm) appelé aussi dans la littérature académique : informatique pervasive, informatique ubiquitaire, informatique omniprésente, propose une vision plus distribuée des entités (capteurs et effecteurs) pouvant participer à l'amélioration des services proposés aux utilisateurs (cas de la domotique par exemple).

L'objectif de l'«Intelligence Ambiante» (IAm) est d'améliorer le confort de l'utilisateur en termes d'assistance ainsi que d'interaction homme-machine. Grâce à la capacité d'un environnement intelligent de percevoir, d'agir et d'interagir pertinemment, par rapport aux besoins de l'utilisateur, les assistants virtuels ont commencé à devenir utiles dans la vie de tous les jours. Un environnement intelligent est composé d'un ensemble d'entités, matérielles et/ou logicielles, intelligentes capables de communiquer et de dialoguer afin de résoudre les besoins issus d'un monde réel et/ou d'un monde virtuel [7]. L'intelligence réside dans la capacité des éléments composants l'environnement de percevoir et de déterminer les besoins de l'environnement et des utilisateurs selon la situation courante.

Nous remarquons que nous répétons fréquemment les mêmes tâches dans nos activités journalières [8], par exemple vérifier la météo avant de sortir, consulter nos agendas, regarder le programme télé, etc. Nous pensons que l'utilisateur préfère déléguer certaines tâches quotidiennes à son environnement intelligent afin d'avoir moins de responsabilités et plus de temps pour d'autres tâches plus confidentielles et plus complexes. Par conséquent, l'utilisateur cherche à satisfaire ses besoins sans aucune intervention explicite de sa part grâce à la capacité de son environnement intelligent de percevoir et d'inférer ses besoins.

Un humain adapte son comportement face aux situations qu'il rencontre. De même, la machine, de manière anthropocentrée, doit effectuer régulièrement la détection de nouveaux contextes, ainsi que l'analyse de situations et la prise de décision optimale.

Dans ce papier, nous nous sommes intéressés à la notion d'adaptativité en interaction homme-machine (IHM). Dans un premier temps, nous montrons que notre système est sensible au contexte, et est capable de notifier l'utilisateur via le canal le plus approprié. Dans un second temps, nous illustrons nos travaux par un scénario d'usage et nous présentons quelques premiers résultats.

2. Etat de l'art

Dans la littérature, le terme "sensibilité au contexte" a été évoqué pour la première fois par [4] dans leurs travaux sur un système de localisation. Ils l'ont défini comme l'aptitude d'une application à s'adapter au contexte de son exécution selon la localisation, l'ensemble des personnes à proximité, les machines, les équipements accessibles, de même que les changements de ces objets dans le temps. Puis, en 2001, Dey l'a défini comme un système proactif qui utilise le contexte pour fournir des informations et/ou des services pertinents (par rapport à la tâche) à l'utilisateur [1]. Schmidt a proposé d'étendre la notion d'un système proactif à la notion d'interaction Homme Machine Implicite (iHCI) où le système recueille des entrées implicites, comme le comportement de l'utilisateur et l'état de son environnement, et propose des services en fonction de ces entrées [3].

L'importance du contexte dans le domaine de l'IHM et les systèmes mobiles a généré des définitions centrées sur l'utilisateur et d'autres sur l'application [11]. Une analyse faite par Brezillon et ses collègues en 2004 concernant les définitions du terme « contexte » les a conduits à conclure que la plupart des définitions sont des réponses aux questions suivantes : qui ? identité ; quoi ? perception et interprétation ; où ? localisation ; quand ? repère temporel ; pourquoi ? compréhension ; comment ? la manière [12].

Aujourd'hui, l'IHM présente des modes d'interaction plus facilement intégrables qu'auparavant (texte, voix, geste, etc.) ainsi que des nouveaux canaux de communication qui ont donné lieu à des réflexions autour d'interfaces multimodales et multicanal plus intuitives et « naturelles » [6].

Dans le cadre des interactions Homme-Machine sur Internet, de nouvelles possibilités émergent depuis peu, grâce à la mise à disposition de puissants outils de surveillance d'événements (météo, flux RSS, tweets, etc.) et de notifications paramétrables, comme IfTTT ou TROPO, par exemple.

IfTTT (If This Then That) [10] est un nouvel outil web qui permet, d'une manière graphique, de faire des "mashups" (des connexions entre des services web, comme par exemple Dropbox, Twitter, Facebook, Youtube, Gmail, etc.). Le principe de IfTTT est de programmer certaines activités en précisant le déclencheur "This" et l'actionneur "That". De manière très intuitive, on peut par exemple programmer de manière graphique la règle "Si je change ma photo de profil Facebook, alors mettre à jour ma photo de profil Twitter."

Dans la même lignée, TROPO [9] est un outil de développement d'applications interactives multiplateforme (SMS, voix) en temps réel. Il intègre le service de communication téléphonique (voix¹, mais aussi émission/réception de SMS), le service d'échange des messages instantanés et il facilite le développement des applications de communication.

Dans nos travaux, nous cherchons également à intégrer les notions de multicanalité et de multimodalité contextuelle mais en insistant sur le dialogue homme-machine. Nous avons choisi l'intégration des deux notions citées ci-dessus afin de rendre l'interaction homme-machine plus naturelle et intuitive.

3. Architecture logicielle

Notre architecture est composée de trois couches (voir Figure 1) qui communiquent et collaborent ensemble afin de mieux répondre aux besoins de l'utilisateur.

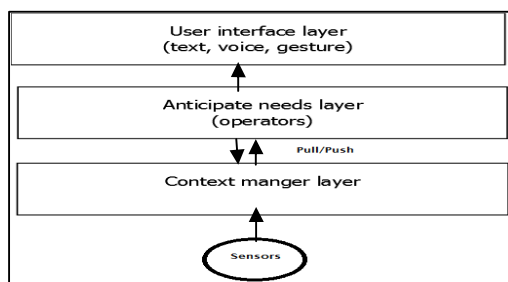


Figure 1. Les différentes couches de l'architecture.

3.1. Couche de Gestion de Contexte

L'environnement intelligent d'un système adaptatif doit assurer une perception continue de l'ensemble de ses capteurs afin de collecter des informations contextuelles en temps réel. Cette mission est gérée par notre première couche appelée "couche de gestion de contexte". Cette couche communique avec un ensemble de capteurs hétérogènes, collecte des informations contextuelles, et les enregistre dans une base de données pour un usage immédiat ou ultérieur [2, 5].

Par exemple, pour l'identification des utilisateurs et le calcul du nombre de personnes présentes dans une pièce, nous utilisons la Radio-Identification (RFID : Radio Frequency Identification) accompagnée d'un tag RFID. Quand un lecteur RFID détecte un tag (voir Figure 2), le système identifie l'utilisateur, le salue (voir Figure 3), et met à jour le nombre de personnes dans la base de données

¹ L'utilisation de la voix est actuellement restreinte à certains pays comme les USA et l'Australie.

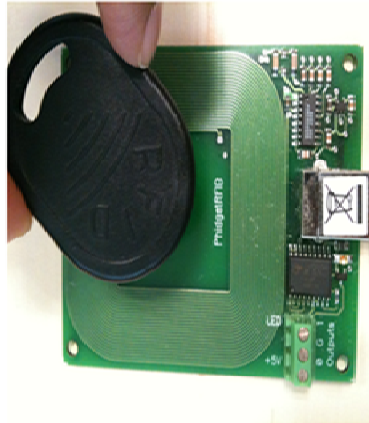


Figure 2. Tag RFID de Bob.

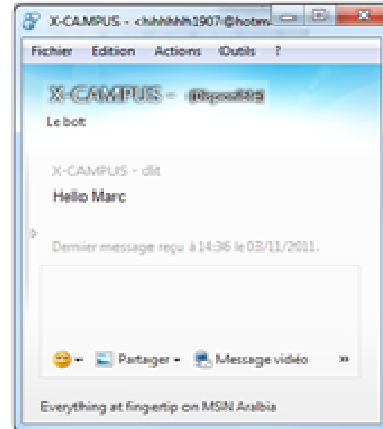


Figure 3. Le système identifie Marc et l'agent le salue

Afin de rendre plus naturelle l'interaction entre l'homme et son environnement, notre système est capable d'initier des conversations avec l'utilisateur, d'une manière proactive et contextuelle. Par exemple, au premier échange de notre système avec l'utilisateur, le système détecte qu'il s'agit d'un nouvel utilisateur ; X-CAMPUS initie alors la conversation et l'invite à faire connaissance via un ensemble des questions (nom, prénom, âge et ville).

Pour des raisons de confidentialité, notre agent peut aussi demander uniquement une autorisation d'accès à une application tierce. Par exemple, X-CAMPUS peut demander l'autorisation de l'utilisateur pour pouvoir accéder à son compte Google Agenda afin de lui notifier ses éventuels rendez-vous.

3.2. Couche d'Anticipation de Besoins

Cette deuxième couche contient une base de règles qu'elle utilise chaque fois qu'une nouvelle situation est détectée. Ces règles sont basées sur le temps, des événements et des données. Par exemple, pour détecter et identifier une personne présente dans tel endroit, le système utilise le lecteur et le tag RFID ainsi que sa base de données. Cette couche communique avec la couche de gestion de contexte en recevant des informations ou en envoyant des requêtes. Pour pouvoir traiter les informations envoyées par notre première couche, nous avons défini trois opérateurs :

Opérateur de conversion : la couche de gestion de contexte enregistre les informations collectées dans leur format initial, puis la couche d'anticipation de besoins convertit ces données brutes en données ayant plus de sens pour le système. Par exemple, quand un capteur de température envoie "25", l'opérateur de conversion interprète cette valeur selon la situation réelle de l'utilisateur par "il fait chaud" ou "il fait froid".

Opérateur d'extraction : dans certains cas, notre système intègre des sources d'informations logicielles comme l'agenda de Google, les flux RSS, etc. Cependant, ces sources peuvent communiquer des informations imprécises. La mission de l'opérateur d'extraction est d'extraire uniquement l'information jugée utile ; par exemple ne garder que les heures et minutes dans une valeur horaire.

Opérateur de couplage : notre système doit, dans certaines situations, agréger un ensemble de données hétérogènes (logique et/ou physique). L'opérateur de couplage collecte des informations de sources différentes afin de satisfaire les besoins dans des situations complexes.

3.3. Couche d'Interaction

Notre travail est basé sur les notions de multimodalité et de multicanalité [6]. La multimodalité permet aux utilisateurs d'employer différents types de modalités (texte, voix, geste, etc.) ; dans notre prototype, nous utilisons le texte, la voix et le geste comme entrée et la voix et le texte comme sortie. La multicanalité permet à l'utilisateur d'interagir avec des canaux différents selon sa situation courante, et assure ainsi la continuité de la communication avec l'utilisateur.

Dans nos travaux, nous utilisons deux canaux différents : le canal téléphonique et le canal Internet. Le but de la couche d'interaction est par conséquent de déterminer la situation de l'utilisateur (statut de l'utilisateur, localisation, nombre et identité des personnes qui l'entourent) afin de choisir le canal le plus approprié et la modalité la plus adéquate. Pour le choix du canal et de modalité, nous distinguons deux types de service différents ; les services temporels, les services événementiels et les services informationnels. Si le contexte perçu par le système implique un service temporel, ce dernier sera confié à un gestionnaire de services afin de pouvoir le déclencher au moment opportun où le choix de type d'interaction sera décidé. Sinon le choix de type de l'interaction se fait au même moment que l'identification de service.

4. Scénario

Afin d'évaluer notre approche, nous avons choisi un scénario qui consiste à proposer, selon l'heure de notification choisie par l'utilisateur ainsi que ses préférences alimentaires, d'aller manger dans le restaurant universitaire le plus approprié.

Dans ce scénario, notre agent nommé X-CAMPUS initialise une phase d'acquisition de connaissances avec l'utilisateur. Il l'invite à s'abonner à son service de restauration en saisissant l'heure de notification souhaitée ; la liste de ses plats préférés ainsi que son numéro de téléphone. Ces dernières informations pourront être mises à jour à tout moment, par messagerie instantanée, en passant par le menu de X-CAMPUS.

Lors de la saisie, notre système effectue quelques vérifications avant l'enregistrement dans sa base de données. Nous distinguons deux types de vérifications : la vérification de format de donnée (HH:MM sur un format 24h pour les horaires de notifications) et la vérification de la cohérence de données. Par exemple, il vérifie si l'utilisateur possède une liste de préférences alimentaires quand il s'est abonné au service de restauration. Notons que la phase d'enrichissement de la base de données de notre système se fait au fur et à mesure des échanges dialogiques qui ont lieu entre l'utilisateur et le système.

Concernant notre deuxième couche d'anticipation de besoins, le déclenchement de service au moment opportun est confié à un module de gestion de services temporels.

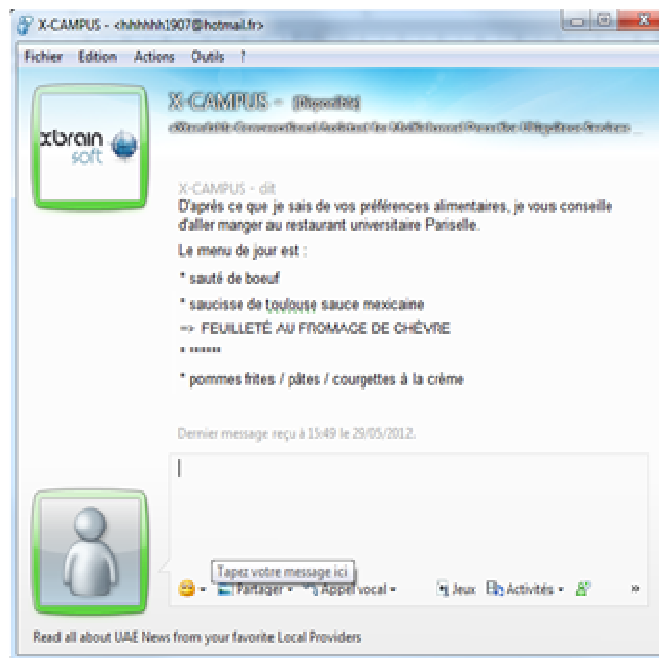


Figure 4. X-CAMPUS-resto via le canal Internet (MSN).

Ce module a besoin de certaines informations pour déterminer la situation de l'utilisateur ; par exemple, il peut vérifier le statut de l'utilisateur (connecté ou déconnecté). Si ce dernier est connecté, X-CAMPUS utilise le canal Internet pour envoyer un message instantané (MI) en affichant en majuscules les préférences alimentaires de l'utilisateur (cf. figure 4). Si l'utilisateur est déconnecté et qu'il a renseigné son numéro de téléphone, un message SMS lui sera envoyé sur son téléphone (cf. figure 5), sinon un e-mail lui sera envoyé de la part de X-CAMPUS.



Figure 5. X-CAMPUS-resto via le canal téléphonique.

Pour tester nos services proactifs, nous avons également développé un simulateur. Ce dernier est une application graphique développée en WPF (Windows Presentation Foundation). Il permet de simuler certaines informations afin de s'affranchir de contraintes informationnelles ou matérielles.

Par exemple, afin de tester à volonté notre système de manière souple, nous pouvons simuler que tel plat sera proposé par tel restaurant, afin d'observer les comportements que cela va engendrer de la part de notre agent. Dans le cas de contraintes matérielles, nous utilisons cette technique pour simuler un ensemble de tags RFID, qui n'existent pas physiquement, et pour lesquels nous pouvons observer les interactions engendrées lors de processus d'identification pour des nombreux utilisateurs.

5. EVALUATION DE TEST

Pour obtenir des retours d'expérience à propos de notre étude, nous avons doté notre agent X-CAMPUS d'un service de satisfaction conversationnel, contextuel et proactif. Ce service demande aux utilisateurs leur niveau de satisfaction pour le service de restauration décrit dans la section précédente. Il s'initie de deux manières différentes selon l'état de connexion de l'utilisateur. Si ce dernier est connecté, X-CAMPUS lui envoie un message instantané sur GTalk ou MSN, sinon il lui envoie un mail contenant l'URL d'un formulaire à remplir en ligne (sur Google docs).

Nous avons demandé aux utilisateurs de donner leur satisfaction par rapport aux critères suivants :

- Respect de l'heure de notification qu'ils avaient programmée ;
- Pertinence du canal choisi par l'agent en fonction du contexte (c'est-à-dire, en l'occurrence, l'état de connexion de l'utilisateur) ;

- Adéquation entre le contenu de menu proposé par X-CAMPUS-resto et les préférences alimentaires de l'utilisateur.

Les utilisateurs étaient invités à exprimer leur niveau de satisfaction en se positionnant sur l'échelle suivante : 1 : *Très insatisfait*, 2 : *Insatisfait*, 3 : *Satisfait*, 4 : *Très satisfait*.

Pour la réalisation d'un test d'expérience, qui s'est déroulée pendant trois jours successifs, nous avons invité quatre utilisateurs à s'abonner à notre service de restauration. Le service de satisfaction globale a été programmé pour se déclencher le troisième jour de notre expérience, selon l'état de connexion de l'utilisateur via un message instantané ou par e-mail. Les résultats obtenus montrent que les utilisateurs sont satisfaits (75%) ou très satisfaits (25%) de X-CAMPUS-resto.

6. Conclusion

Dans ce papier, nous avons montré que notre agent X-CAMPUS est capable de fournir aux utilisateurs un service proactif contextuel multicanal nommé X-CAMPUS-resto. La proactivité contextuelle de notre agent se vérifie par sa capacité à anticiper les besoins issus du monde réel à travers la perception qu'il en a ; la multicanalité de notre agent se vérifie quant à elle par sa capacité à basculer d'un canal à un autre selon le contexte calculé pour chaque utilisateur.

Nous avons mis en place une première expérience de test autour d'un service de restauration universitaire afin d'étudier le comportement du système en nous focalisant sur les contraintes de proactivité contextuelle et de multicanalité. Les résultats obtenus ont montré que les participants sont globalement satisfaits de notre agent X-CAMPUS-resto et qu'ils estiment que le choix du canal (messagerie instantanée, SMS ou e-mail) est à chaque fois justifié en fonction de leur état de connexion. Après cette première expérience concluante, nous allons étendre cette expérience auprès de nombreux utilisateurs, et intégrer la notion de la multimodalité.

7. REFERENCES

- [1] A.K. Dey. « Understanding and using context ». *Personal and ubiquitous computing*, vol. 5, p. 4-7. 2001.
- [2] A. K. Dey, D.M. Salber, G.D. Abowd. An architecture to support context-aware applications. *UIST' 99*.
- [3] A. Schmidt. Interactive Context-Aware systems interacting with ambient intelligence, in G. Riva, F. Vatalaro, F. Davide & M. Alcaniz (eds), *Ambient Intelligence*, IOS Press, chapter 9, p. 159-178. 2005.
- [4] B. N. Schilit, N. Adams et R. Want. «Context-Aware Computing Applications ». In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA. p. 85-90. Santa Cruz, CA,: IEEE Computer Society. 1994.
- [5] J. Hong. The context fabric: an infrastructure for context-aware computing. *CHI 2002*, Minneapolis, Minnesota, USA. ACM 1-58113-454-1/02/0004. 2002.
- [6] J. Rouillard. *Adaptation en contexte : contribution aux interfaces multimodales et multicanal*, Habilitation à diriger des recherches, 2008.
- [7] H. Sassi. *Comment l'intelligence ambiante peut-elle nous assister ? Informatique des Organisations et Systèmes d'Information et de Décision*, INFORSID, Lille, 2011.
- [8] H. Sassi, J. Rouillard. *Proactive Assistance Within Ambient Environment*, *Intelligent Systems and Applications*, pp. 60-65, INTELLI, Chamonix, 2012.
- [9] <https://www.tropo.com>
- [10] <http://ifttt.com>
- [11] M. Miraoui. *Architecture logicielle pour l'informatique diffuse : modélisation du contexte et adaptation dynamiques des services*, Thèse de doctorat, Montréal, 2010.
- [12] P. Brezillon, M. R. Borges, J.A. Pino et J.-Ch. Pomerol. « Context-Awareness in Group Work: Three Case Studies ». In *2004 IFIP Int. Conf. on Decision Support Systems (DSS 2004)*. Prato, Italie. 2004.