



PREDIT 2001-04
MINISTERE DE L'EQUIPEMENT, DES TRANSPORTS ET DU LOGEMENT
DIRECTION DES TRANSPORTS TERRESTRES

Dossier DTT 216-75-01 – Décision de subvention du 31 déc. 2004

MOUVER.PERSO
MObilité et mUltimodalité Voyageurs Etudiants
en Région Nord Pas de Calais
—
Système d'information multimodale personnalisée
Rapport final - Décembre 2006

Rédacteurs du rapport :

E. Grislin-Le Strugeon, A. Anli (LAMIH)

En relation avec :

C. Petit-Rozé, G. Uster (INRETS),
O. Walbecq et M. Zidi (ARCHIMED),
M. Abed, G. Conreur, P. Dos Santos et C. Kolski (LAMIH)



Université de Valenciennes et du Hainaut-Cambrésis
Le Mont Houy, F-59313 Valenciennes cedex 9
Fax : 03.27.51.13.16



Table des Matières

1	Introduction.....	3
1.1	Dates.....	3
1.2	Intervenants.....	3
1.3	Objectif.....	3
1.4	Echéancier des travaux.....	4
2	Définition et création des services	6
2.1	Objectifs.....	6
2.2	Méthodologie	7
2.3	Développement du système d'information.....	8
2.3.1	Analyse des besoins	8
2.3.2	Conception du service.....	10
2.3.3	Implémentation du service	14
2.4	Développement du SMA.....	16
2.4.1	Analyse des modèles d'agents	16
2.4.2	Conception des compétences	17
2.4.3	Implémentation des compétences	20
2.4.4	Intégration	23
3	Evaluations, analyse et documentation	26
3.1	Tests techniques	26
3.2	Test fonctionnels	27
3.3	Evaluation de la personnalisation	28
3.3.1	Vérification de l'adaptation	29
3.3.2	Vérification de l'adaptation au profil de l'utilisateur.....	31
3.3.3	Vérification de l'adaptation du profil d'un utilisateur selon l'historique des interactions ..	32
3.3.4	Vérification de l'adaptation de la réponse selon les événements	32
3.3.5	Vérification de la qualité de la personnalisation.....	33
3.4	Evaluation de performance	37
4	Perspectives.....	39
5	Conclusion	40
6	Références.....	41
7	Annexe	43

1 Introduction

Ce rapport présente le travail effectué dans le cadre du projet PREDIT intitulé « MOUVER.PERSO : MObilité et mUltimodalité Voyageurs Etudiants en Région Nord Pas de Calais – Système d'information multimodale personnalisée) » .

1.1 Dates

Date de notification de l'arrêté par le Ministère de l'Équipement, des Transports et du Logement : 31 décembre 2004, projet prévu sur une durée de 24 mois.

1.2 Intervenants

Les personnes suivantes participent à ce projet :

Équipe d'encadrement : Christophe KOLSKI (professeur), Mourad ABED (professeur) et Emmanuelle GRISLIN-LE STRUGEON (Maître de Conférences) ;

Ingénieur de recherche : Christelle PETIT-ROZÉ ;

Ingénieurs d'étude : Gérald CONREUR, Philippe DOS SANTOS ;

Doctorant : Abdouroihamane ANLI ;

Partenaire : société Archimed (Lille) représentée par Mongi ZIDI et Olivier WALBECQ.

Ces acteurs ont travaillé en relation avec Guillaume USTER (INRETS-ESTAS, Villeneuve d'Ascq).

Nous avons aussi fait travailler sur ce projet plusieurs étudiants de niveau Bac+4 et Bac+5, en projet ou stage en laboratoire de recherche.

1.3 Objectif

Ce projet intitulé MOUVER.PERSO est la deuxième phase du projet MOUVER (Mobilité et Multimodalité Voyageurs Etudiants en Région Nord Pas de Calais). La première phase, financée par la Prédim – Direction des Transports Terrestres en 2003, a permis de mettre en avant des pistes de travail pour l'amélioration de l'information de mobilité à destination d'une population de jeunes et d'étudiants.

Cette deuxième phase est consacrée à la mise en oeuvre concrète des services imaginés et à leur évaluation. La personnalisation de l'information en est l'axe central.

Les études conduites lors de la première phase du projet MOUVER ont mis en évidence [Marzloff 03a][Marzloff 03b] un ensemble de directions à suivre dans un objectif d'amélioration des dispositifs d'information transport en particulier collectif, à destination d'une population estudiantine.

Le choix de cette population se justifie par le fait que :

1. c'est une population qui n'a pas toujours accès à la voiture et qui trouve ce mode trop onéreux à l'usage,

2. c'est une population sensible à l'écologie et au développement durable,
3. c'est une population qui représente le monde de la mobilité de demain,
4. c'est une population intéressée par les nouvelles technologies de l'information et de la communication.

Parmi les pistes de travail proposées dans le domaine de l'information multimodale, nous nous intéressons ici particulièrement à « l'articulation entre l'individuel et le collectif », à savoir l'adaptation de l'information fournie aux besoins spécifiques de l'utilisateur. Lors de cette seconde phase du projet MOUVER, nous souhaitons mettre en œuvre et évaluer des services liés à la personnalisation des informations. Ces services seraient centrés sur la **mobilité au quotidien**, dans un but d'accompagnement du jeune dans son « univers » et ses activités.

Plus précisément, notre objectif est de développer un démonstrateur, désigné par « MOUVER.PERSO », capable de :

- assurer une passerelle entre le monde de la mobilité et la société de l'information ;
- extraire des gisements de données transports de l'information pertinente pour les usagers ;
- organiser la mobilité quotidienne en fonction de l'agenda et des offres de transport afin de gagner du temps et mieux organiser le déplacement de l'étudiant ;
- présenter l'information de manière immédiate et naturelle. Seront distinguées l'information statique et l'information temps réel ;
- diffuser l'information sur plusieurs supports, tout en assurant une présentation ergonomique et un contenu homogène.

Cette démarche sera menée dans le cadre d'une veille technologique : recensement des nouveaux outils, concepts, langages, standards et normes, dans le domaine de l'information déplacement. Un accent sera plus particulièrement mis sur l'approche « logiciel libre ».

1.4 Echancier des travaux

Le projet se déroule durant la période : janvier 2005 à décembre 2006.

Le projet se décompose en deux phases. La première phase se focalise sur la définition et la création de services, la seconde a pour objet leur déploiement et leur évaluation. Le travail s'est réparti de la manière suivante :

Phase 1 (janvier 2005 à décembre 2005) :

- *Etape 1 (3 mois) : **Expression des besoins et spécification.*** L'expression des besoins pourra se faire sous forme de cas d'utilisation. S'agissant d'une démarche de prototypage, les besoins ne seront pas figés, et ils seront présentés de manière large, avec éventuellement des variantes, qui seront précisées en fonction des possibilités d'implémentation qui seront identifiées lors de la conception du démonstrateur. Ensuite, il s'agira de proposer des spécifications du démonstrateur, qui le décrivent en termes fonctionnels (services), indépendants des solutions techniques qui seront retenues.
- *Etape 2 (9 mois) : **Conception et implémentation.*** On procèdera à la conception, qui décrit les choix techniques, et les justifie par rapport à d'autres variantes envisageables, puis à l'implémentation, qui consiste à développer le logiciel et à décrire les problèmes pratiques rencontrés.

- *En parallèle et en fin d'étape 2, l'étape 3 (4 mois) : Préparation des tests.* Il s'agit de proposer le protocole expérimental considérant un ensemble de tests, à mettre en œuvre dans l'étape 4.

Phase 2 (janvier 2006 à décembre 2006) :

- *Etape 4 (10 mois) : Tests et améliorations.* Les tests et les améliorations successives concernant les services apportés et les modes d'interaction entre le système et les sujets (usagers des TC).
- *Etape 5 (2 mois) : Documentation du démonstrateur et analyse des résultats.* Il s'agit de rédiger la documentation, qui explique comment utiliser le démonstrateur. On procède aussi à l'analyse globale des résultats et la mise en évidence des pistes à creuser pour les futurs développements

2 Définition et création des services

Nota : Cette partie reprend en partie le contenu du rapport intermédiaire [ref à mettre], daté de Janvier 2006.

La première phase du projet concernait la définition et la création des services, consistant à analyser et définir précisément les services visés et développer une première version des applications permettant de mettre en œuvre ces services.

L'innovation principale du système d'information multimodale et personnalisée consiste à aider à l'organisation de la journée en terme de boucles de mobilité quotidienne, en proposant le meilleur choix modal et/ou le meilleur enchaînement des tâches contenus dans les agendas personnels et professionnels. Nous traitons ici l'univers de l'étudiant et par ce biais observons le monde des études, des loisirs, de la famille, des achats, etc., mais également du travail. En effet, le recours à des jobs à temps partiel est de plus en plus fréquent pour le financement des études.

Le système a pour mission d'inciter à l'utilisation des transports en commun ou des modes doux en présentant des solutions de déplacement personnalisées. La sensibilisation à la mobilité durable sera omniprésente. Personnaliser les informations consiste à présenter certaines informations à bon escient, au bon moment, de façon ciblée pour l'utilisateur. Il s'agit de prendre en compte ses contraintes, besoins et préférences dès lors qu'un choix peut être fait parmi un ensemble de possibilités de modes de transports et d'horaires.

2.1 Objectifs

La définition des besoins est basée sur les objectifs définis dans le plan de recherche.

L'objectif général de ce projet est d'assurer une passerelle entre le monde de la mobilité et la société de l'information. Cela consiste alors à mettre en œuvre des services d'accompagnement personnalisés autour de la mobilité quotidienne des étudiants.

Les étudiants disposent d'un emploi du temps initialement "statique" (mais qui évolue au cours du temps). Autour de cet emploi du temps, ils occupent leurs temps libres de différentes manières : sports et loisirs, shopping, études, culture...

L'originalité de ce projet est d'accompagner les étudiants tout au long de leurs occupations de manière personnalisée :

- en terme de déplacements (moyen de transport préféré, coût du déplacement, . . .)
- en terme d'organisation des journées (par rapport aux préférences des étudiants et en fonction de l'offre de transport).

L'emploi du temps des étudiants¹ devient alors "dynamique".

Par définition, un agenda est un support prédaté sur lequel on peut noter les choses à faire ou déjà faites. Les agendas ont diverses présentations ; ils prennent, par exemple, la forme d'un carnet contenant une page pour chaque jour, semaine ou mois. Un agenda électronique est un logiciel de

¹ Le terme "emploi du temps" est à prendre ici au sens large. Il comprend l'emploi du temps en tant qu'étudiants mais aussi les autres activités planifiées. Dans la suite du document il sera remplacé par le terme agenda.

gestion du temps qui facilite la planification horaire, quotidienne ou à plus long terme, de l'utilisateur.

Un agenda personnel peut alors être défini comme étant un outil personnalisé de gestion du temps d'un utilisateur donné.

Il se présente comme tout agenda, mais sa particularité réside dans ses fonctionnalités.

Ainsi il doit permettre à l'utilisateur de

1. visualiser le contenu de son agenda de manière journalière, hebdomadaire...
2. ajouter ou supprimer des rendez-vous via une interface de saisie.

Mais, un agenda personnel, tel que nous l'envisageons, doit posséder des fonctionnalités supplémentaires :

3. organiser (ou ré-organiser) dynamiquement la planification horaire des RDV en fonction de l'utilisateur
4. gérer les déplacements entre les différents RDV
5. être accessible à distance
6. utiliser éventuellement des données de géolocalisation
7. s'interfacer éventuellement avec d'autres agendas

Parmi ces points d'étude, nous nous sommes principalement intéressé à la gestion des déplacements entre rendez-vous, ce qui constitue l'originalité des travaux dans le cadre de ce projet.

2.2 Méthodologie

Notre approche est guidée par la méthodologie *PerMet (Personalization Methodology)* (Cf. figure 1), développée au LAMIH dans le cadre de nos travaux sur la conception de systèmes multi-agents pour la personnalisation des informations [Anli 05a][Anli 05c][Petit-Rozé 04]. De cette façon, les différentes étapes fournies par la méthode, ainsi que leur enchaînement, dirigent la réalisation du présent projet. En retour, leur application nous permet de préciser et de conforter leur description et leur positionnement.

La méthode repose sur un principe de séparation du système d'information (SI) et du système multi-agent (PerSyst) procurant le service de personnalisation. Leur conception fait donc l'objet de phases séparées (les 2 branches du "Y" en figure 1), avant leur intégration.

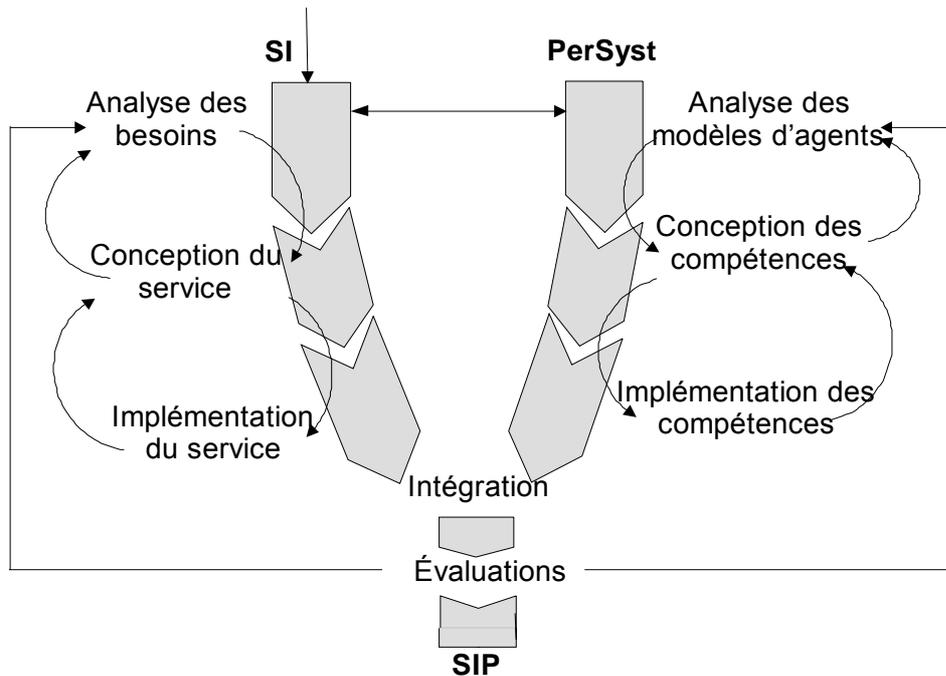


Figure 1: Méthodologie PerMet

En suivant le séquençement proposé par la méthode PerMet, nous débuterons donc par la description de la conception du système d'information.

2.3 Développement du système d'information

Le point d'entrée fourni par la méthode est l'analyse des besoins en terme de service.

2.3.1 Analyse des besoins

Lors de cette première étape, il s'agit d'effectuer une analyse pour prendre en compte les besoins des utilisateurs en terme de services personnalisés. Cette analyse va piloter l'étape d'analyse des modèles d'agents dans la phase PerSyst. C'est aussi dans cette étape qu'on spécifie le modèle de données (DTD) qui seront échangées entre le SI et PerSyst.

Nous avons détaillé l'objectif et les fonctionnalités associées :

- fournir une information transport pertinente pour l'usager
- planifier des itinéraires
- gérer l'agenda personnel
- diffuser l'information en assurant l'ergonomie de l'IHM
- gérer la présentation de l'information sur différents supports

Les fonctionnalités mises en évidence sont représentées en figure 2.

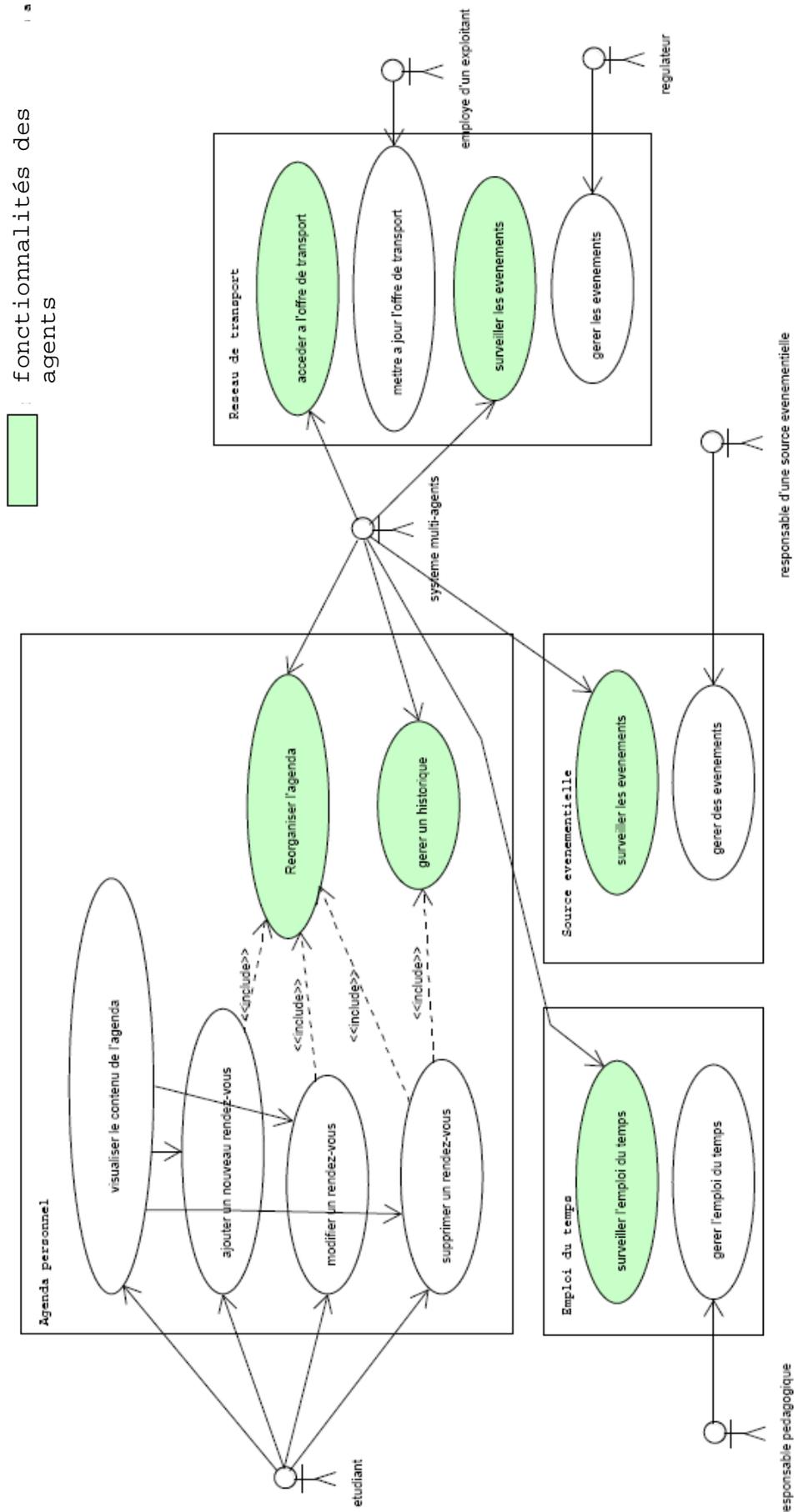


Figure 2: Diagramme UML de cas d'utilisation décrivant les principales fonctionnalités d'un agenda personnel

2.3.2 Conception du service

Lors de cette étape, il s'agit de concevoir le service à personnaliser (application externe) sur la base de l'analyse effectuée. L'architecture, les choix techniques, la conception de l'interface utilisateur, etc., sont réalisés à ce niveau.

Chaque cas d'utilisation du diagramme présenté en figure 2 a donné lieu à la réalisation du diagramme d'activité du service correspondant.

Visualiser son agenda

Visualiser le contenu de son agenda est une fonctionnalité de base. Elle consiste à résumer une semaine ou une journée, ou détailler un rendez-vous. Il est possible de passer du résumé d'une semaine à celui d'une journée, ou encore d'une journée au détail d'un rendez-vous comme cela est illustré par le diagramme d'activité visible en figure 3.

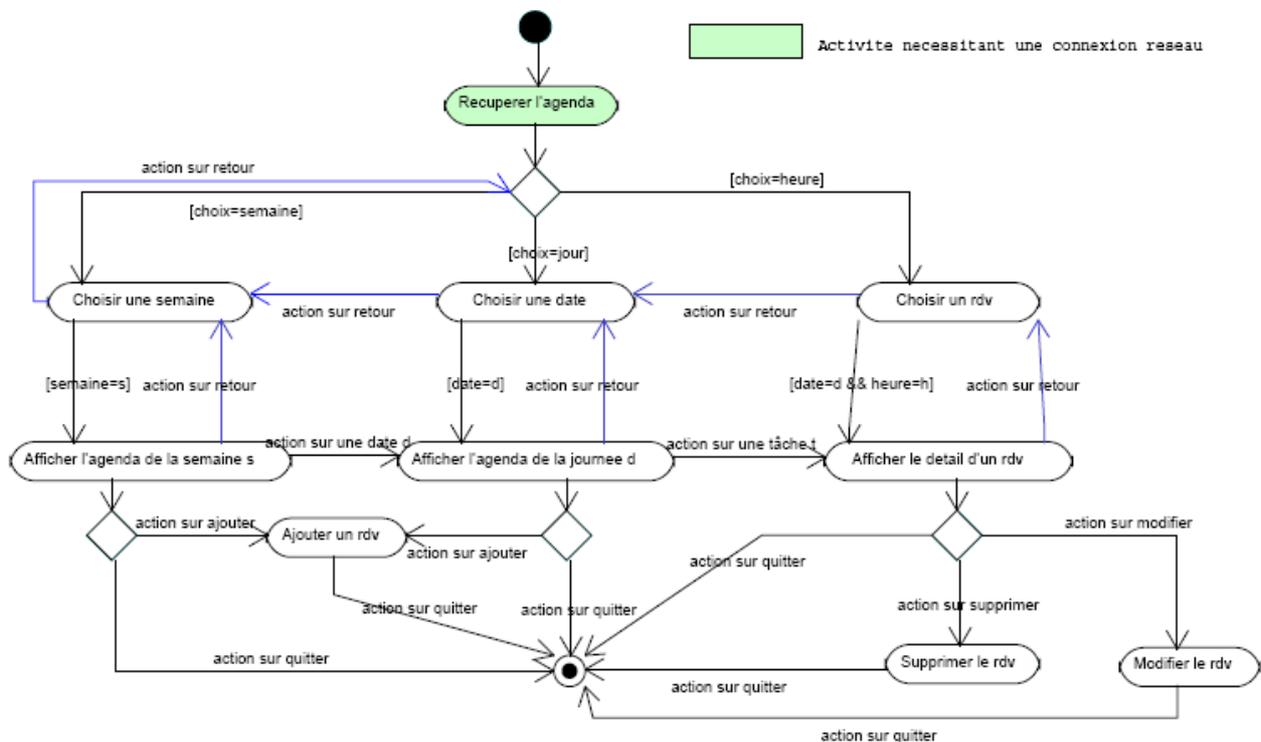


Figure 3: Diagramme d'activité décrivant la fonctionnalité "visualiser le contenu de son agenda".

Définition de la fonctionnalité :

- Étape 1 : Récupérer le contenu de l'agenda de l'étudiant. Cette étape sera dépendante de la mémoire du pc, pda. . . avec lequel interagit l'étudiant.
- Étape 2 : Sélectionner un niveau de visualisation (semaine, journée, rendez-vous).
- Étape 3 : L'étudiant a la possibilité de naviguer entre les différents niveaux de visualisation et peut faire appel à d'autres fonctionnalités telles que ajouter, modifier ou supprimer un rendez-vous.

Supprimer un rendez-vous

Supprimer un rendez-vous est une fonctionnalité de base que nous proposons d'améliorer. Elle consiste à libérer le créneau occupé par le rendez-vous et mettre à jour les déplacements du rendez-vous précédent et du rendez-vous suivant comme cela est décrit par le diagramme d'activité visible en figure 4.

Définition de la fonctionnalité :

- Étape 1 : Sélectionner le rendez-vous à supprimer.
- Étape 2 : Mettre à jour l'agenda en fonction de l'offre transport.
- Étape 3 : Ré-organiser la journée en comblant le créneau libre.

Les étapes 2 et 3 nécessitent l'intervention d'un agent logiciel situé au niveau du serveur. Ces étapes sont rassemblées dans l'activité Supprimer(date,heureDeb) nécessitant une connexion réseau.

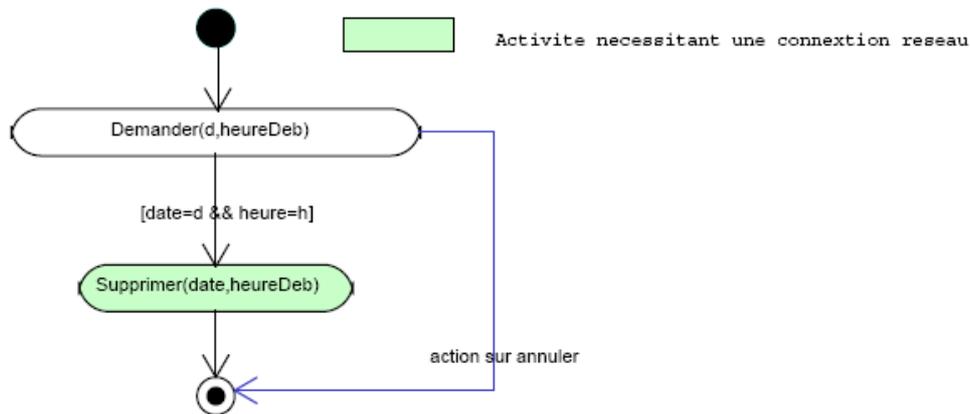


Figure 4: Diagramme d'activité décrivant la fonctionnalité "supprimer un rendez-vous".

Ajouter un rendez-vous

Ajouter un rendez-vous est une autre fonctionnalité de base que nous proposons d'améliorer. Le but de l'amélioration est d'optimiser l'ajout du rendez-vous.

L'utilisateur peut alors

1. imposer la date et l'heure de début. Ce type d'ajout est qualifié de complètement spécifié (CS).
2. imposer uniquement la date mais pas l'heure de début. Ce type d'ajout est qualifié de spécifié (S).
3. demander à placer un rendez-vous. Ce type d'ajout est qualifié de non spécifié (NS).
4. prévoir un rendez-vous à une date donnée. Ce dernier cas n'est pas considéré comme un ajout mais une planification.

Ces différents cas sont pris en compte dans le diagramme d'activité visible en figure 5.

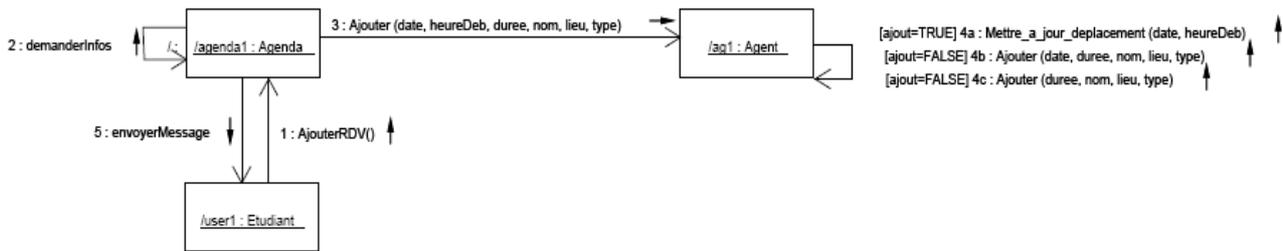


Figure 6: Diagramme de collaboration décrivant les interactions nécessaires pour "ajouter un rendez-vous".

Organiser (ou ré-organiser) son agenda

La réorganisation se fait en fonction de l'utilisateur (prenant ainsi en compte les événements importants ou non pour ce dernier) et de l'offre de transport.

L'organisation (ou la réorganisation) de l'agenda est déclenchée par l'une des quatre actions suivantes :

- ajouter un rendez-vous,
- supprimer un rendez-vous,
- modifier un rendez-vous ou
- modifier une donnée transport.

Le schéma fourni en figure 7 décrit la réorganisation suite à l'une des quatre actions ci-dessus exprimées par l'utilisateur.

Comme cela a été dit précédemment, la gestion de l'agenda dépend de l'offre de transport mais dépend aussi des choix de l'étudiant. Afin de prendre en compte les choix de l'étudiant, nous avons défini des types de rendez-vous afin de gérer des priorités entre ceux-ci. Par exemple, un RDV de type *examen* sera par défaut prioritaire sur tous les autres types de RDV.

Ces types pourront évoluer, et de nouveaux types pourront s'ajouter au fur et à mesure des besoins, par inférence ou demande explicite de l'étudiant. Nous envisageons également de permettre à l'utilisateur de définir ses propres priorités entre types de RDV.

Par défaut, trois types sont prédéfinis. Ils sont décrits selon leurs caractéristiques : *nom*, *descriptif* et *règle de priorité*, dans le tableau 1. Dans ce tableau, *LType* représente la liste des types définis dans l'agenda et la fonction *priorite(t)* fournit une valeur correspondant à l'importance du type *t* pour l'étudiant.

Nom	Descriptif	Règles de priorité
examen	Type relatif à un examen.	$\forall t \in LType, \text{priorite}(t) < \text{priorite}(\text{examen})$
job	Type relatif à un job d'étudiant.	$\forall t \in LType, t \neq \text{examen}, \text{priorite}(t) < \text{priorite}(\text{job})$
default	Type défini par défaut.	$\forall t \in LType,$ $\text{priorite}(\text{default}) \begin{cases} < \text{priorite}(t) \text{ si } t = \text{job} \vee t = \text{examen} \vee \\ & (t = \text{default} \wedge \text{default.importance} < t.importance) \\ = \text{priorite}(t) \text{ si } t = \text{default} \wedge \\ & \text{default.importance} = t.importance \\ > \text{priorite}(t) \text{ si } t = \text{default} \wedge \\ & \text{default.importance} > t.importance \end{cases}$

Tableau 1: Définition des types de rendez-vous.

2.3.3 Implémentation du service

C'est l'étape de la réalisation effective du service. Le service est développé conformément aux modèles conceptuels définis lors de la phase de conception du service. Des tests du service peuvent s'effectuer par simulation des données à fournir au service.

Concernant les fonctionnalités standard aux agendas, nous avons fait le choix de l'utilisation d'un agenda déjà existant, qui est celui fourni par le portail web développé par la société Archimed partenaire du projet. L'authentification est gérée par un annuaire LDAP (Lightweight Directory Access Protocol, protocole d'annuaire standardisé permettant le partage d'informations pour des applications via le réseau, Cf. [IETF]), conformément aux normes et usages actuels. Nous utilisons des transformations XSL (langage de description standardisé, Cf. [12]) pour le rendu graphique.

Concernant le format de données, nous avons travaillé sur une extension du schéma XML de l'agenda standard, afin d'intégrer les données d'itinéraire dans le planning. Ceci était nécessaire pour interfacer la maquette avec le système de personnalisation qui gère les données. C'est pour cela que nous avons été amené à étudier les formats de stockage de données des agendas existants. Nous avons retenu des formats ouverts et basés sur XML qui est le format d'échange de données entre les agents du système. Suite à cette étude, nous avons retenu xCalendar [NWG 05]. C'est un format basé sur le langage XML qui permet l'ajout de nouveaux tags en fonction de nos besoins tels que le descriptif des itinéraires permettant le passage d'un rendez-vous à un autre. Un exemple de fichiers xCalendar est donné en figure 7.

```

<?xml version="1.0" encoding="UTF-8" ?>
<iCalendar>
<vcalendar>
<prodid>-//Mozilla.org/NONSGML Mozilla Calendar V 1.0 //EN</prodid>
<version>2.0</version>
<vevent>
<uid>13b2cfe0-eedc-11d9-a48d-c624b0692935</uid>
<summary>éRunion Mouver-Perso</summary>
<dtstamp>20050707T093356Z</dtstamp>
<dtstart>20050708T080000Z</dtstart>
<dtend>20050708T120000Z</dtend>
<description>Note de la érunion</description>
<categories>
<item>Projets</item>
</categories>
<location>LAMIH</location>
<url>http://www.univ-valenciennes.fr/LAMIH</url>
<rrule>:FREQ=WEEKLY;INTERNAL=1;BYDAY=FR</rrule>
<valarm>
<trigger>-PT10M</trigger>
</valarm>
</vevent>
</vcalendar>
</iCalendar>

```

Figure 7: Un rendez-vous au format xCalendar.

Les interfaces spécifiques aux PDA sont en cours de finition.

Cette étape terminant la phase spécifique au développement du système d'information, nous détaillons maintenant la phase de développement du SMA (branche de droite dans la représentation de la méthode PerMet, Cf. figure 1).

2.4 Développement du SMA

2.4.1 Analyse des modèles d'agents

La première étape de cette deuxième phase est destinée à spécifier les différents modèles d'agents utiles pour les besoins de personnalisation. Ce travail est basé sur les besoins en terme de service exprimés dans l'étape d'analyse du système d'information.

Le résultat de cette étape est une description des fonctionnalités que doivent fournir les différents modèles d'agents et les modèles d'interaction entre les différents agents.

Il s'est agi essentiellement ici d'étendre l'ensemble de modèles et l'architecture du SMA que nous avons développé lors de travaux antérieurs, en particulier pour le projet AgenPerso².

L'architecture résultante se décompose en cinq parties (Cf. figure 8 et également référence [Anli 05b]) correspondant à des modèles d'agents logiciels : coordination, gestion des profils, recherche d'information, gestion des agendas, assistance à l'utilisateur.

Chaque partie est assurée par un ou plusieurs agents aux rôles différents et aux compétences correspondantes :

- les agents d'assistance s'exécutent sur les plates-formes d'interaction des utilisateurs. Un agent de ce type est capable de reconnaître la plate-forme à partir de laquelle l'étudiant s'est connecté et par laquelle les informations doivent s'afficher.
- les agents de coordination assurent la distribution des tâches et l'intégration des résultats fournis par les autres types d'agents.
- les agents de gestion de profil créent et mettent à jour les informations que le système possède sur les utilisateurs afin d'assurer la personnalisation des résultats.
- les agents de recherche d'information sont situés sur des serveurs d'application et/ou sur des serveurs d'information.
- enfin, la gestion des agendas est assurée par des agents dotés de compétences nécessaires à la gestion dynamique de l'agenda des utilisateurs.

Les compétences sont définies à partir des fonctionnalités mises en évidence dans la partie 2.3.

² Projet PREDIT, 2001-03.

Figure 8: Architecture du démonstrateur MOUVER.PERSO

Nous avons conçu ce système de manière à assurer les caractéristiques suivantes :

- multi-applications : le SP peut communiquer avec plusieurs applications pour assurer la prise en compte des différents modes d'interaction et des différentes plate-formes accueillant l'interaction.
- autonomie et pro-activité : l'assistance de l'utilisateur, la recherche d'information, la gestion des profils et des agendas exigent des capacités d'autonomie et de pro-activité.
- support de mobilité : la mobilité consiste à faire migrer un code informatique d'une plate-forme vers une autre pour s'y exécuter. Les différents éléments du SP peuvent ne pas tous se situer sur une même machine. Par exemple, la gestion de profil utilisateur peut se situer sur un serveur LDAP [Rizcallah 00] alors que la recherche d'information peut se situer sur un serveur d'information. L'administrateur du SP peut déployer ces éléments à partir d'une seule interface d'administration.
- évolutivité : l'administrateur du SP peut ajouter des éléments, en enlever, faire évoluer leur comportement. Par exemple, permettre à un module de recherche d'information transport de faire aussi de la recherche d'information culturelle, etc.

2.4.2 Conception des compétences

Chaque modèle d'agent décrit dans l'étape précédente est ensuite détaillé afin de concevoir l'ensemble des compétences qui lui sont nécessaires pour assurer les tâches qui lui sont attribuées.

Par exemple, l'agent de gestion des profils doit posséder les compétences lui permettant d'accéder à l'annuaire LDAP des utilisateurs, d'enregistrer le choix de l'utilisateur, de personnaliser les informations des perturbations, d'envoyer éventuellement des courriels, de personnaliser les itinéraires, etc.

Chacune de ces compétences a ensuite été détaillée. Par exemple, la compétence *KsocialLastFilteringSkill* de personnalisation d'itinéraire suit le diagramme d'activités présenté en figure 9. La méthode utilisée privilégie la réutilisation de résultats précédents :

- l'agent commence par rechercher le profil de l'utilisateur, et dans l'historique associé à celui-ci, l'existence de cette même requête. Si cette recherche aboutit, le dernier choix d'itinéraire effectué est repris.
- Si le profil existe mais qu'aucun résultat associé ne peut être repris, l'agent recherche des utilisateurs au profil similaire sur lesquels baser la sélection de l'itinéraire.
- Enfin, si l'utilisateur n'est pas connu (son profil n'existe pas dans le système), la sélection se base sur l'ensemble des choix effectués par les utilisateurs selon le principe du vote majoritaire.

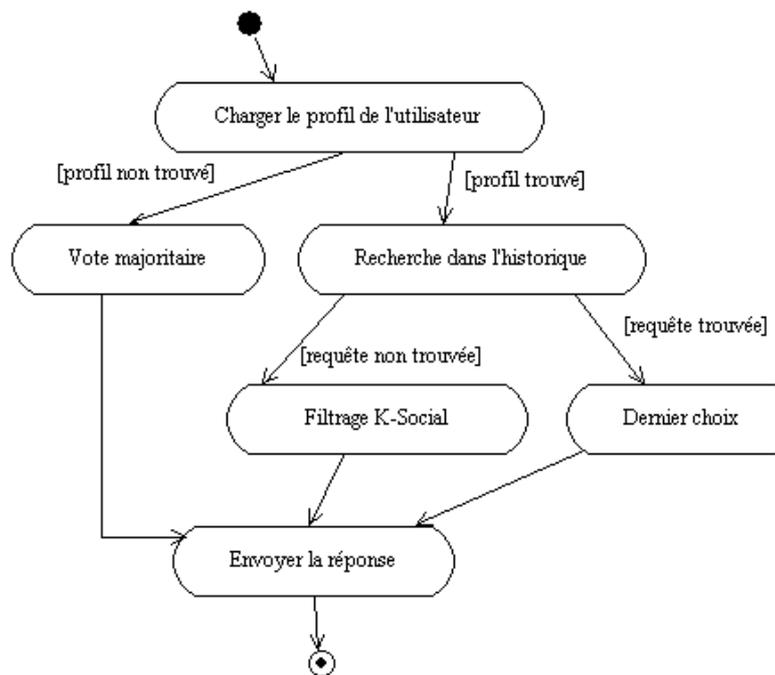


Figure 9: Représentation générale de l'activité de personnalisation d'itinéraire des agents de gestion de profil.

Chacune des sous-méthodes représentées par les activités ci-dessus, est ensuite détaillée. Par exemple, le filtrage *K-Social* correspond au fonctionnement décrit en figure 10 : l'agent vérifie, pour chacun des utilisateurs connus, s'il a déjà réalisé la requête présente. Si c'est le cas, il calcule le degré de similarité entre cet autre utilisateur et l'utilisateur présent. Dans le cas où ce degré de

similarité (Sim) est supérieur à un seuil donné (K), le choix de cet autre utilisateur est pris en compte afin de déterminer la recommandation d'itinéraire qui sera fournie par l'agent.

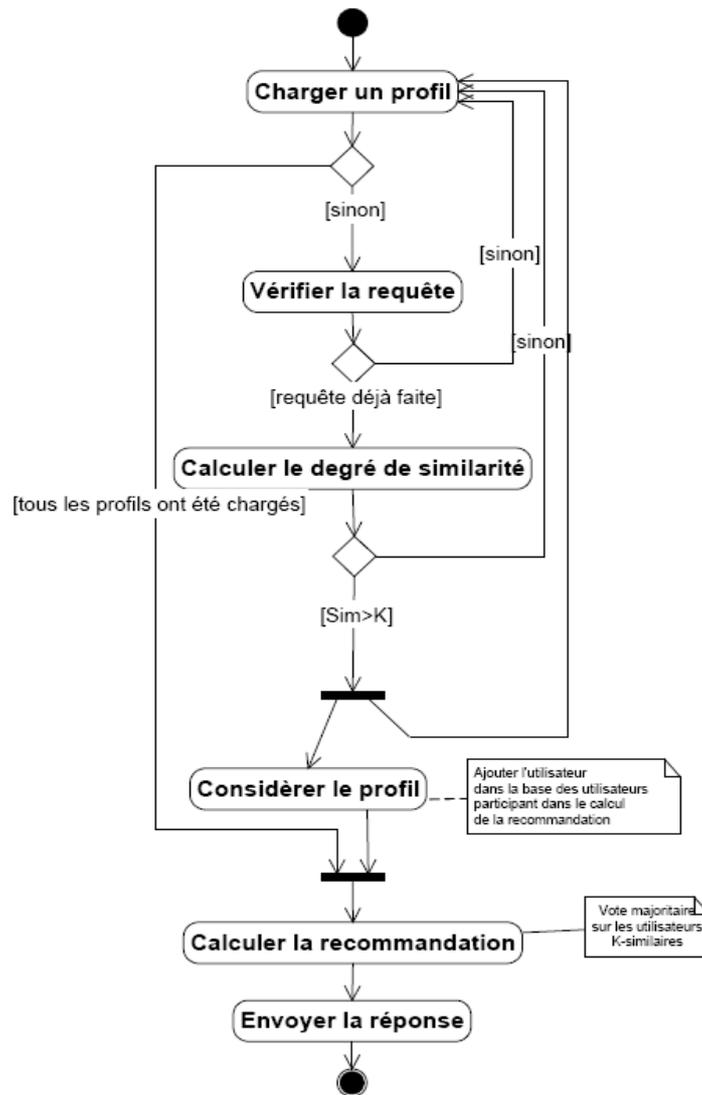


Figure 10: Détail du filtrage K -Social.

Une autre compétence des agents de gestion de profil est la mise à jour des connaissances par apprentissage. Nous avons déjà effectué une étude sur ce point [Petit-Rozé 05b] ; dans le cadre de ce projet, l'activité d'apprentissage des agents de profil a été conçue selon le schéma donné en figure 11.

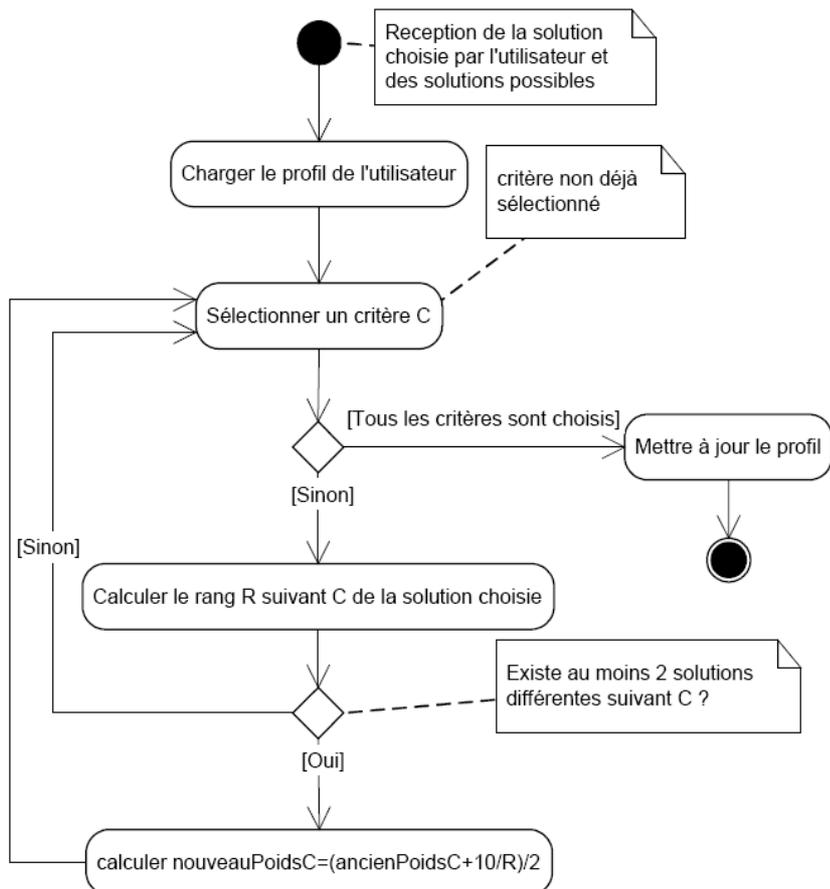


Figure 11: Mise à jour du profil de l'utilisateur.

2.4.3 Implémentation des compétences

Il s'agit dans cette étape de la réalisation effective des compétences.

Par exemple, concernant les agents de gestion de profils, le format de stockage des données constituant un profil a été défini (Cf. figure 12) sur la base de travaux précédents [Petit-Rozé 05a]. Il comporte trois parties : une partie statique comportant les données qui ne sont pas amenées à évoluer, ou très peu, telles les données personnelles de l'utilisateur ; une partie dynamique, formée des données sur lesquelles va s'appliquer l'apprentissage ; une partie historique, destinée à stocker les requêtes et choix précédents de l'utilisateur.

```

<ENTRY>
<Static><DN>LDAP://SRV-ETCL-DC01:389/CN=Anli,OU=Sitp,DC=DOMTRANSPORT,DC=local <DN> </Static>

<TransportPreferences>
  <MoinsCorrespondance>6 </MoinsCorrespondance>
  <MoinsTemps>8 </MoinsTemps >
  <MoinsCouteux>7</ MoinsCouteux >
  <MoinsMarche>5</MoinsMarche>
</TransportPreferences>

<Historic>
  <TransportHistoric>
    <Request>
      <LieuDepart>LAMIH</LieuDepart>
      <LieuArrivee>Archimed</LieuArrivee>
      <Date>17/01/2006</Date>
      .....
    </Request>
    <Result>
      <Trajet>
        <Depart>LAMIH</Depart>
        <Arrivee>Gare de Valenciennes</Arrivee>
        <Mode>Bus</Mode>
        ....
      </Trajet>
      +<Trajet>
      .....
  
```

$$\mathcal{M}_u = \mathcal{S}_u \cup \mathcal{P}_u \cup \mathcal{H}_u$$

\mathcal{S}_u : Partie Statique (Annuaire LDAP)
 \mathcal{P}_u :Partie Pondérée (Préférences)
 \mathcal{H}_u :Historique
 $\mathcal{H}_u = \bigcup_{i \in I} (R_i \cdot C_i^u)$

Figure 12: Détail d'un profil utilisateur.

Pour exemple, nous présentons en figure 13 le diagramme de classes correspondant à l'implementation des compétences des agents de gestion de profil :

- la compétence *KsocialLastFilteringSkill*, de personnalisation des itinéraires en fonction des modèles d'activité fournis par l'étape de conception des compétences ;
- la compétence *LDAPSkill*, d'accès au service d'annuaire LDAP pour obtenir les données personnelles de l'utilisateur, par exemple un numéro de téléphone afin de l'avertir d'une perturbation par un SMS pendant son déplacement en utilisant la compétence complémentaire *DisturbanceSkill* ;
- la compétence *ManipulateProfileSkill*, permettant à l'utilisateur de gérer son profil (visualisation, suppression ou modification) ;
- la compétence *SaveChoiceSkill*, d'enregistrement du choix effectué par l'utilisateur ;
- les compétences *DisturbanceInfoSkill* et *DisturbanceSkill*, de personnalisation de l'information de perturbation ;
- la compétence *MailSkill*, permettant l'envoi de courriel.

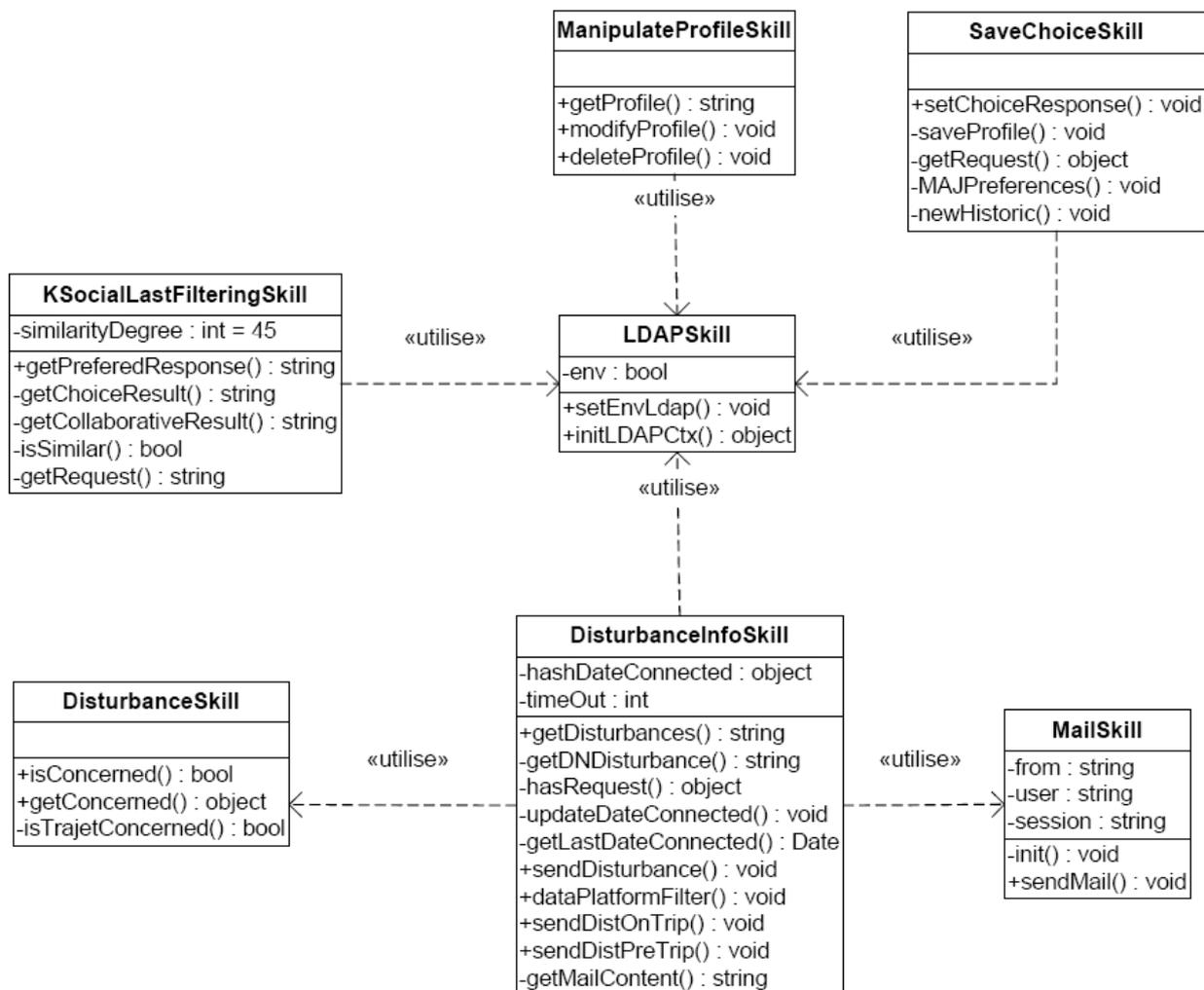


Figure 13: Diagramme des classes développées pour les agents de gestion de profil.

La compétence *KsocialLastFilteringSkill* inclut l'algorithme central de personnalisation. La méthode utilisée consiste à :

1. rechercher le profil de l'utilisateur
2. si le profil n'existe pas, la méthode de vote majoritaire est utilisée : la réponse donnée correspond à celle choisie par la majorité des utilisateurs.
3. si le profil existe, la requête est recherchée dans l'historique de l'utilisateur.
 - si la requête n'est pas trouvée, la méthode de filtrage collaboratif est utilisée : la réponse correspond à celle choisie par la majorité des utilisateurs ayant un profil similaire au sien
 - sinon, c'est le dernier choix qui est retenu : la dernière réponse proposée pour cette requête à cet utilisateur est fournie de nouveau.

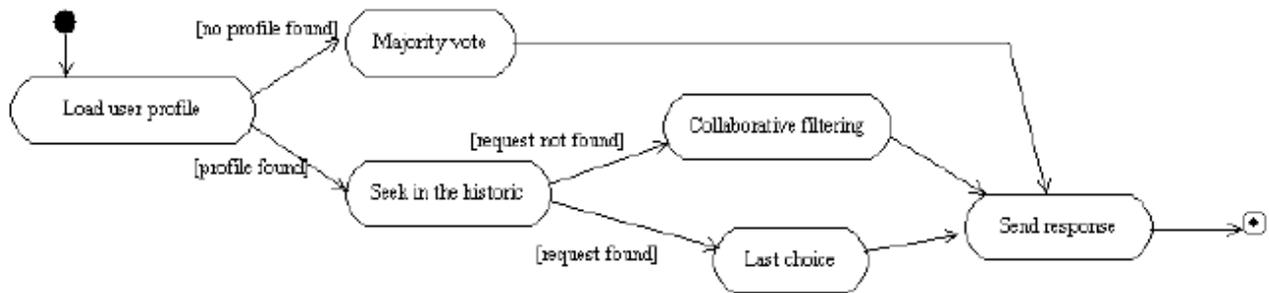


Figure 14: Méthode de personnalisation utilisée.

2.4.4 Intégration

C'est l'étape d'intégration de SI et du SMA pour former le système global. Concrètement, il s'agit d'utiliser les recommandations fournies par le SMA afin d'adapter le comportement du SI.

Nous avons choisi d'utiliser le format SOAP pour les échanges entre les différents éléments du système. SOAP (Simple Object Access Protocol) est un protocole d'échange entre applications, indépendant de toute plate-forme, et basé sur le langage XML. Il s'agit d'appels de services encadrés par des balises XML et transmis selon le protocole HTTP. Nous nous en sommes servis en tant qu'élément d'interface entre l'agenda et le SMA (fig. 14).

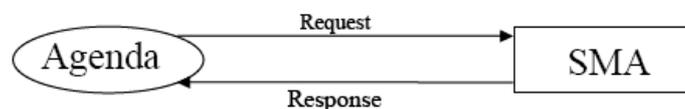


Figure 15: Echange d'informations entre l'agenda et le SMA.

Deux méthodes de communication ont été prévues :

- une méthode synchrone, qui nécessite une attente de réponse de la part des interlocuteurs ;
- une méthode asynchrone, qui correspond à un simple échange de données sans nécessité de blocage de l'exécution des interlocuteurs.

Pour exemple, détaillons la méthode synchrone utilisée dans le cadre d'une modification de l'agenda entraînant une recherche d'itinéraire adapté aux préférences et contraintes de l'utilisateur. Ceci se concrétise par l'appel suivant de la part de l'agenda à l'agent de communication *Com* :

```
Response = Com.request("searchItinerary",Request)
```

L'ajout d'un nouveau RDV dans l'agenda va se concrétiser par la création d'un fichier XML comportant les informations de dates, lieux, etc., qui sera complété par le SMA en fonction des

informations à propos des autres RDV présents dans l'agenda et à propos des préférences connues de l'utilisateur. En figure 15, par exemple, le lieu de départ est déduit par le système.

Request :

```

<ENTRY>
<DN>LDAP://SRV-ETCL
DC01:389/cn=test,OU=Sitp,DC=DOMTRANSP
RT,DC=local</DN>
<Request>
<LieuDepart>Lille_Flandres</LieuDepart>
<HeureArrivee>08:15</HeureArrivee>
<date>25/01/2006</date>
<LieuArrivee>Lille_Europe</LieuArrivee>
</Request>
</ENTRY>

```

Figure 16: Le SMA complète les informations fournies par l'utilisateur sur la base des données qu'il possède

Le SMA effectue la recherche d'itinéraire demandée, laquelle produit plusieurs solutions (balise "Result" dans le fichier XML de la figure 16). Grâce aux données qu'il possède sur l'utilisateur, les solutions possibles sont ordonnées, de celle qui est supposée la préférée jusqu'à celle supposée la moins intéressante pour cet utilisateur. La réponse fournie en tant que résultat de l'appel ci-dessus est constituée du fichier XML contenant les solutions ordonnées. Ce fichier est lu et interprété pour fournir les résultats présentés à l'utilisateur (Cf. fig. 16).

Response :

```
<ENTRY>
<DN>...</DN>
+<Request>
<Response>
+<Result>
+<Result>
+<Result>
+<Result>
<Trajet>
<Depart>Lille_Flandres</Depart>
<Arrivee>Lille_Europe</Arrivee>
<TC>Marche</TC>
<Heure_Depart>08:00</Heure_Depart>
<Heure_Arrivee>08:10</Heure_Arrivee>
<Prix>0</Prix>
<Duree>10</Duree>
<Plan>http://...</Plan>
</Trajet>
+<Critere>
</Result>
</Response>
<Choice>
<Result>4</Result>
<Reason>LAST</Reason>
<Value></value>
</Choice>
</ENTRY>
```

Heure	Mode	Durée	Prix	Exploitant
Lille_Flandres 08:00	Marche	10 min	0 euros	Plan d'accès
Lille_Europe 08:10				

Nombre de correspondance : 1
Durée totale : 10.0
Coût total : 0.0 euros

Le SP vous recommande cet itinéraire car vous l'avez préféré, la dernière fois

Imprimer | Modifier

Légende

-  Fragment ajouté par l'agent de recherche d'itinéraire
-  Fragment ajouté par l'agent de gestion de profil

Figure 17: La réponse du SMA est utilisée par le SI afin de proposer une solution adaptée aux besoins et préférences de l'utilisateur.

Concernant le gestionnaire d'agenda oueb, nous avons étudié les solutions existantes. Il nous est apparu qu'aucune d'elles ne répondait à l'ensemble de nos critères, à savoir : une solution adaptée aux PDA, gratuite, autonome (non hébergée) et open source. De même, concernant les bibliothèques de développement, il existe des bibliothèques de lecture mais aucune ne répond à nos besoins en écriture. C'est pourquoi nous avons opté pour un développement spécifique, seule solution viable en réponse à nos besoins.

Les interfaces côté PDA ont été développées.

En conclusion de cette partie, notons que cette étape constitue le point auquel nous sommes parvenus fin décembre 2005.

3 Evaluations, analyse et documentation

3.1 Tests techniques

Les tests techniques étaient destinés à détecter d'éventuels défauts introduits lors de l'implémentation du système.

Les tests unitaires ont porté sur les 2 parties de l'application développée, à savoir le système d'information (SI) et le système de personnalisation (SP).

Au niveau du SI, nous avons testé les différents modules et fonctions développés pour l'agenda oueb et pour l'agenda PDA.

- L'agenda oueb étant basé sur l'agenda fourni par la plate-forme MASC d'Archimed, nous avons testé seulement les nouveaux modules que nous avons ajoutés pour prendre en compte le transport et l'aspect personnalisé des itinéraires à fournir. Par exemple, nous avons testé les fonctions développées pour le parsing XML des événements stockés au niveau de MASC afin d'extraire les éléments nécessaires (lieu d'arrivée, date d'arrivée, etc.) pour la recherche des itinéraires correspondants. Nous avons aussi testé les fonctions de gestion des erreurs comme, par exemple, lorsque le système de personnalisation est indisponible (suite à une panne du serveur de personnalisation, etc.).
- Les tests unitaires de l'agenda PDA ont surtout porté les fonctions de parsing des documents XML. Nous avons vérifié que les modules PHP fonctionnent aussi bien sur une plate-forme Windows que sur une plate-forme Linux. Les erreurs mises en évidence par ces tests ont été corrigées, notamment, un dysfonctionnement d'une fonction de parsing des itinéraires lorsque le système était déployé sur une plate-forme Windows (le système est initialement développé sous Linux et la fonction ne présentait pas d'erreur) : sous Windows, la fonction ne renvoyait pas correctement les itinéraires composés de plus de deux trajets (alors que sous Linux tout semblait correct).

Au niveau du SP, les tests ont porté principalement sur les algorithmes de filtrage pour la personnalisation des itinéraires. Les différentes activités (diagrammes d'activités définis dans la phase de conception de compétences cf. 2.4.2) possibles des agents ont été vérifiées. Nous avons, par exemple, vérifié que l'algorithme de filtrage social implémenté fournit bien le résultat calculé manuellement. Des tests sur les algorithmes d'adaptation automatique du SMA ont aussi été effectués. Par exemple, nous avons instancié deux agents de gestion de profil dans deux plates-formes différentes et avons vérifié la redirection automatique des messages à un agent lorsque l'une des plates-formes est fermée (simulation d'une panne ou d'un dysfonctionnement d'un agent).

Les tests d'intégration ont porté sur les interactions entre le SI et le SP, et le SIP dans sa globalité. Ces tests concernaient l'interopérabilité des implémentations du protocole d'interaction SOAP utilisé par l'agenda oueb (implémentation ASP), l'agenda PDA (implémentation PHP) et le système de personnalisation (implémentation Java). Nous avons remarqué que l'échange direct d'objets (ou de type natif, entier, double, etc.) Java, PHP ou ASP posait des problèmes de conversions de type. C'est pour cela que nous avons opté pour l'échange des données sous forme de chaînes de caractères.

Le test du SIP dans sa globalité n'a pas révélé de problème particulier. Les utilisateurs de l'agenda oueb sont reconnus au niveau de l'agenda du PDA et inversement. La mise à jour des préférences utilisateur sont repercutées dans l'ensemble du SIP (agenda oueb, agenda PDA, SP).

3.2 Test fonctionnels

Les tests fonctionnels sont destinés à vérifier que le système répond aux fonctionnalités attendues, définies par les cas d'utilisation. Ces tests ont été effectués sur l'agenda oueb et sur l'agenda PDA. Les tests de l'agenda oueb ont été effectués sous Windows XP avec le navigateur Internet Explorer version 6.0 et avec le navigateur FireFox version 1.5. Les tests de l'agenda PDA ont été effectués sous Microsoft Windows Mobile avec le navigateur Internet Explorer Mobile.

Tests fonctionnels de l'agenda oueb

<i>Cas d'utilisation</i>	<i>tests effectués</i>	<i>résultats - commentaires</i>
Visualiser le contenu de l'agenda	-Visualisation journalière -Visualisation hebdomadaire -Visualisation d'un événement	OK. Se base sur l'agenda MASC.
Ajouter un nouveau RDV	-RDV en remplissant tous les champs (objet, date, etc.) -RDV sans objet	-OK pour IE 6.0 Pour le RDV sans objet, l'utilisateur est invité à remplir l'objet par un message. -Echec pour FireFox 1.5. Ceci est dû à des scripts implémentés par MASC non conforme au standard W3C.
Modifier un RDV	-Modification de tous les champs (objet, date, etc.) -Modification d'un seul champ (date)	OK. Se base sur l'agenda MASC.
Supprimer un RDV	-Suppression d'un RDV -Suppression de tous les RDV	OK. Se base sur l'agenda MASC.
Gérer l'historique	-Visualisation des préférences -Modification des poids associés aux critères -Suppression du profil (préférences+historique)	OK. L'utilisateur ne peut pas modifier les données de l'historique mais il peut le supprimer (en supprimant son profil).
Surveiller les événements	-Perturbation sur une ligne d'un exploitant -Perturbation sur plusieurs lignes d'un exploitant -Perturbation sur plusieurs lignes de différents exploitants	OK. Envoi de mail. Affichage sur le portail oueb. Simulation d'envoi de SMS
Accéder à l'offre transport	-Visualisation d'un itinéraire associé à un RDV -Choix d'un autre itinéraire	OK

Tests fonctionnels de l'agenda PDA

<i>Cas d'utilisation</i>	<i>tests effectués</i>	<i>résultats - commentaires</i>
Visualiser le contenu de l'agenda	-Visualisation journalière -Visualisation hebdomadaire -Visualisation d'un événement	OK.
Ajouter un nouveau RDV	-RDV en remplissant tous les champs (objet, date, etc.) -RDV sans lieu de départ, d'arrivée ou sans date	OK. Pour le RDV sans lieu de départ, d'arrivée ou sans date, l'utilisateur est invité à remplir ces champs par un message.
Modifier un RDV	-Modification de tous les champs (objet, date, etc.) -Modification d'un seul champ (date)	OK.
Supprimer un RDV	-Suppression d'un RDV -Suppression de tous les RDV	OK.
Gérer l'historique		Utilisation de l'agenda oueb
Surveiller les événements		Utilisation de l'agenda oueb
Accéder à l'offre transport	-Visualisation d'un itinéraire associé à un RDV -Choix d'un autre itinéraire	OK. Utilisation de l'agenda oueb pour le choix d'un autre itinéraire

3.3 Evaluation de la personnalisation

Les évaluations de la personnalisation consistent à évaluer la qualité de personnalisation réalisée. Il s'agit de vérifier si la personnalisation effectuée au niveau du service répond bien aux préférences de l'utilisateur.

Les illustrations proposées proviennent d'un simulateur de PDA sur PC pour des raisons de lisibilité du rapport, les saisies d'images écran de PDA étant moins claires. La figure xx présente la même image, en a) sur PDA et en b) sur le simulateur.

FIGURE A AJOUTER

3.3.1 Vérification de l'adaptation

Ces tests ont pour objectif de vérifier que la réponse fournie par le SIP tient compte des choix précédents des utilisateurs, même dans le cas d'une utilisation en mode "anonyme" : l'utilisateur n'est pas inscrit, son profil est donc inconnu.

Scénario :

L'utilisateur est Marie, une étudiante à l'université de Valenciennes, au Mont Houy. Elle réside le week-end chez ses parents à Douai et loue une chambre étudiante à Aulnoy pour la semaine.

Elle utilise le système en mode "anonyme". Elle ajoute à son agenda son activité de retour de Douai à sa résidence étudiante pour ce dimanche soir (cf. fig. 18).

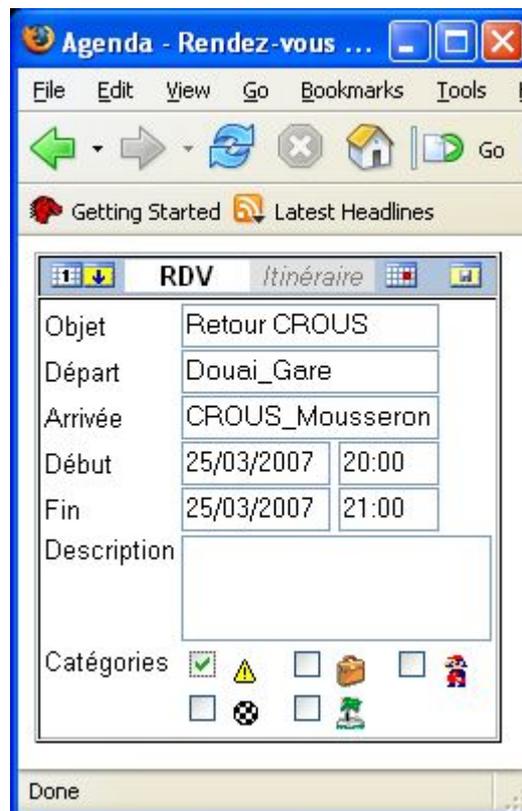


Figure 18: Ecran de saisie d'un rendez-vous sur l'agenda du PDA.

Les icônes proposées par l'agenda lors de l'ajout d'un événement sont :

<i>Icône</i>	<i>Usage prévu</i>
	événement important (ex. : examen)
	événement de type "cours"
	événement lié au travail
	événement à caractère sportif
	événement de loisir

La barre de titre de la fenêtre contient les boutons permettant de naviguer dans l'agenda, et un bouton *Itinéraire*, grisé lors de la saisie d'un nouvel événement. Ce bouton devient actif après l'ajout, à savoir, dès lors qu'un itinéraire est proposé par le système en rapport avec l'événement.

L'utilisateur qui consulte son agenda peut ainsi accéder à l'itinéraire proposé lorsqu'il sélectionne un événement enregistré sur son agenda (cf. fig. 19).

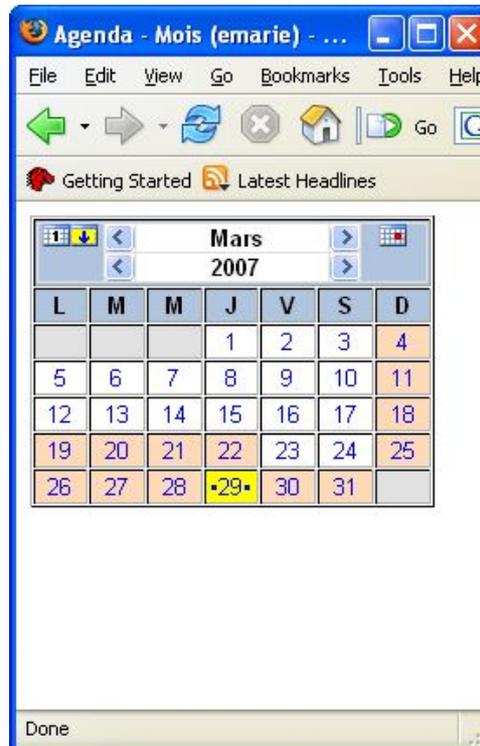
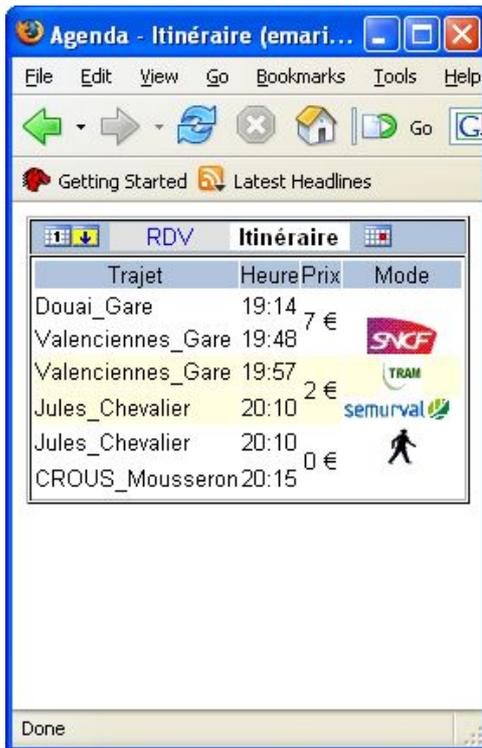


Figure 19: Visualisation de l'agenda après ajout d'un événement le 29 mars.

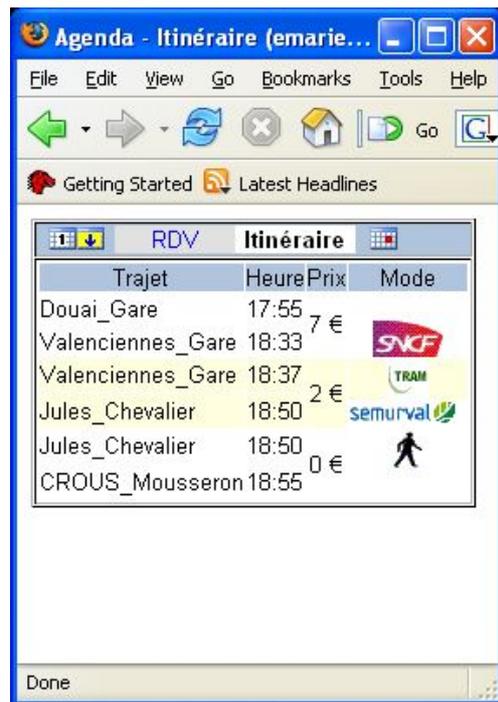
L'utilisateur étant inconnu du système, la personnalisation ne peut s'appliquer en fonction du profil de la personne. Le système peut dans ce cas avoir différents comportements. Nous avons choisi de proposer les réponses suivantes (mais ceci est paramétrable) :

- dans le cas où une requête similaire a déjà été réalisée par d'autres utilisateurs, le système propose la solution majoritairement enregistrée pour les utilisateurs inscrits ayant effectué cette même requête (illustration du résultat fourni en fig. 20-a) ;
- dans le cas contraire, le système propose la solution arrivant en première lieu dans la liste des itinéraires possibles (illustration du résultat fourni en fig. 20-b).

La différence entre les deux réponses montre l'utilisation qui est faite de l'historique des itinéraires calculés. Les solutions proposées se différencient ici par les horaires fournis, toutefois les différences peuvent, bien entendu, porter également sur le choix des moyens de transport utilisés et leur chaînage.



a)



b)

Figure 20: Ecrans de visualisation des itinéraires proposés suite à l'ajout de l'événement dans l'agenda : a) si la requête existe dans l'historique, prise en compte des solutions retenues pour les autres utilisateurs, b) si la requête n'a jamais été formulée auparavant.

3.3.2 Vérification de l'adaptation au profil de l'utilisateur

Ces tests ont pour objectif de vérifier que la réponse fournie par le SIP s'adapte en fonction du profil de l'utilisateur.

Scénario : inscription au système / création de profil par défaut / adaptation de la réponse (filtrage selon vote majoritaire sur la classe d'utilisateurs ayant un profil identique à celui du nouvel utilisateur)

Trois nouveaux utilisateurs s'inscrivent au système via la page ouébe. Ils donnent entre autres les informations suivantes :

- le premier est étudiant à l'université de Valenciennes et habite Douai.
 - le second est un enseignant qui habite Villeneuve d'Ascq. Son emploi du temps montre qu'il commence plusieurs fois par semaine à 8h et termine souvent après 19h.
 - le troisième est un enseignant qui habite Somain.
- création du nouvel utilisateur dans l'annuaire LDAP
 - création du profil par défaut selon les critères suivants : ...
 - première proposition d'itinéraire selon les données enregistrées : ...

3.3.3 Vérification de l'adaptation du profil d'un utilisateur selon l'historique des interactions

Ces deuxièmes tests ont pour objectif de vérifier que le système est capable d'adapter le profil de l'utilisateur en fonction de l'historique des interactions...

Scénario : *adaptation d'un profil selon l'historique des interactions (filtrage selon le dernier choix)*

Un étudiant a l'habitude de venir en train jusqu'à la gare de Valenciennes, puis de se rendre en tramway à l'université. Le système lui propose donc en priorité ce choix de déplacement.

Une analyse médicale récente ayant montré qu'il a un taux trop élevé de cholestérol, son médecin lui recommande de marcher un peu plus. L'étudiant décide alors de choisir désormais d'aller en train jusqu'à la gare de Trith le Poirier, avant de finir le reste du trajet à pied. Il va refuser une fois l'itinéraire en train jusque Valenciennes suivi du tramway ; ensuite le système lui proposera en priorité le train jusque le Poirier avec le reste à la marche.

Scénario : *adaptation de la réponse selon l'historique des interactions (filtrage K-social)*

Un enseignant doit se rendre exceptionnellement à Douai pour une visite de stage. Il n'a jamais fait cette requête au système, mais d'autres personnes ayant le même profil que lui l'ont fait. C'est l'itinéraire majoritairement choisi par ces autres utilisateurs qui lui ressemblent qui sera proposé par le système.

Scénario : *adaptation du profil manuellement*

Une étudiante possédant un profil dans le système, se casse une jambe. Elle peut changer son profil manuellement via la page ouébe afin que le système ne lui propose plus de trajet comportant de la marche.

3.3.4 Vérification de l'adaptation de la réponse selon les événements

Ces troisièmes tests ont pour objectif de ...

vérifier que le système adapte la réponse lorsqu'il y a entrée d'un nouvel événement du type

- nouveau RDV
- annulation
- modification d'horaire
- perturbation transport

Scénario : ajout d'un événement / réorganisation des déplacements / prise en compte d'une perturbation

étudiant est informé de l'ouverture de la billetterie pour un concert qui l'intéresse à Paris. Il souhaite aller au centre-ville de Valenciennes acheter les places. Or, il possède une plage libre d'1h30 dans son agenda aujourd'hui. Il ajoute l'événement à son agenda qui lui propose un itinéraire en tramway. Il se rend par ce moyen à Valenciennes, fait son achat et s'apprête à repartir par le même moyen lorsque le système l'informe d'une perturbation sur la ligne de tramway, et lui propose une solution alternative en bus pour le retour.

3.3.5 Vérification de la qualité de la personnalisation

Les évaluations de personnalisation ont porté sur une étude de la qualité, a priori, de la personnalisation fournie à l'utilisateur. Il s'agit d'une étude qualitative des méthodes pour la personnalisation de l'information transport. Les méthodes étudiées concernaient deux méthodes de filtrage collaboratif (la méthode de similarité des vecteurs et la méthode de Pearson ; voir chapitre 1, §1.1.4.2 du projet de Master Recherche [Boidin 05]).

Pour pouvoir évaluer les méthodes d'apprentissage automatique, deux bases de données utilisateurs tests ont été créées. La première base de données est une base de référence et la seconde est une base pour les prédictions. Les bases de données générées sont représentées sous forme matricielle : une matrice pour les rangs des critères par rapport aux solutions et ce pour toutes les requêtes et une matrice de choix des itinéraires pour chaque utilisateur considéré et sur l'ensemble des requêtes. Elles sont, par ailleurs, composées de deux sous-matrices :

- l'une de taille (nbTotalSolutions x nbCritères)
où nbTotalSolutions est le nombre total de solutions
$$nbTotalSolutions = \sum_{j=1}^{nbRequêtes} nbSolutions_j$$

nbRequêtes est le nombre de requêtes,
nbSolutions_j est le nombre de solutions pour la requête j,
nbCritères est le nombre de critères pris en compte (ici nbCritères=4) ;
- et l'autre de taille (nbUsers x nbRequêtes)
où nbUsers correspond au nombre d'utilisateurs de la base de données
et nbRequêtes est le nombre de requêtes.

Génération de la base de données de référence

Cette première base de données correspond à une situation dans laquelle tous les utilisateurs ont passé l'ensemble des requêtes et pour chacune d'elles, ont choisi un itinéraire, en se basant sur un scénario. Nous avons fait l'hypothèse qu'un utilisateur choisit un itinéraire en fonction d'un seul critère et que l'ensemble de ses choix sera fonction de ce critère.

Les étapes de l'algorithme de création de la base de données de référence sont les suivantes :

1. Fixer le nombre d'utilisateurs,
2. Fixer le nombre de requêtes,
3. Pour chaque requête, fixer aléatoirement un nombre de solution entre 1 et 10,
4. Générer aléatoirement les rangs des critères par rapport aux solutions, de telle manière que les solutions soient triées vis-à-vis des critères,
5. Générer aléatoirement, pour chaque utilisateur, un critère de sélection d'itinéraire,
6. Pour l'ensemble des utilisateurs et pour chaque requête, choisir la solution qui répond à ce critère, l'historique de chaque utilisateur est ainsi obtenu (figure 17).

	R1	R2	R3	S2	S3	S4
	S1 (1,1,1,1)	S1 (1,1,1,1)	S1 (3,2,1,2)	(4,1,4,4)	(1,4,2,3)	(2,3,3,1)
User1	X	X			X	
User2	X	X	X			
User3	X	X			X	
User4	X	X		X		
User5	X	X	X			
User6	X	X				X
User7	X	X			X	
User8	X	X				X
User9	X	X		X		
User10	X	X				X

Figure 21 : Exemple de base de données utilisateurs de référence (10 utilisateurs et 3 requêtes)

Génération de la base de données utilisateurs pour les prédictions

Cette base de données, basée sur la première, sert pour le calcul des prédictions. Pour cela, contrairement à la précédente où tous les utilisateurs ont choisi une solution pour chacune des requêtes, cette base n'est remplie que partiellement, aux deux tiers pour être précis. Elle a été créée à partir d'une copie de la précédente, puis un tiers des solutions a été retiré.

Les étapes de l'algorithme de création de cette base de données sont :

1. Copier la base de données utilisateurs de référence,
2. Choisir aléatoirement, pour chaque utilisateur, un tiers des requêtes dont on enlèvera la solution choisie,
3. Retirer la solution choisie (le choix passe à 0), pour l'ensemble des utilisateurs, un nouvel historique est ainsi obtenu (figure 18).

	R1	R2	R3	S2	S3	S4
	S1 (1,1,1,1)	S1 (1,1,1,1)	S1 (3,2,1,2)	(4,1,4,4)	(1,4,2,3)	(2,3,3,1)
User1	X	--	--	--	--	--
User2	--	X	X			
User3	X	--	--	--	--	--
User4	X	X		X		
User5	--	X	--	--	--	--
User6	--	--				X
User7	X	X			X	
User8	--	X	--	--	--	--
User9	X	X		X		
User10	X	--				X

Figure 22 : Exemple d'historiques incomplets des utilisateurs

Cette matrice sera par la suite complétée avec les prédictions résultant du filtrage collaboratif.

Génération des préférences des utilisateurs

Les rangs des critères par rapport aux solutions étant définis et les choix des utilisateurs pour certaines requêtes étant effectués (figure 18), le système peut alors générer les notes associées aux critères pour chaque utilisateur. Les notes sont alors stockées dans une matrice de taille (nbUsers x nbCritères) où nbUsers est le nombre d'utilisateurs et nbCritères le nombre de critères considérés, (ici, nbCritères=4) (figure 19).

	C1	C2	C3	C4
User1	10,000	5,569	5,858	5,792
User2	6,060	6,097	10,000	7,278
User3	10,000	5,569	5,858	5,792
User4	5,583	10,000	6,268	6,208
User5	6,060	6,097	10,000	7,278
User6	5,500	6,510	7,143	10,000
User7	10,000	5,569	5,858	5,792
User8	5,500	6,510	7,143	10,000
User9	5,583	10,000	6,268	6,208
User10	5,500	6,510	7,143	10,000

Figure 23 : Exemple de préférences des utilisateurs

Algorithme d'évaluation

Une fois les prédictions faites pour le tiers des requêtes non passées par l'ensemble des utilisateurs, nous allons vérifier si les prédictions réalisées correspondent effectivement à ce qui était prévu. Pour cela, nous avons procédé suivant l'algorithme décrit ci-après.

Les étapes de cet algorithme sont les suivantes :

1. Compter le nombre de prédictions fausses en comparant la matrice de référence et la matrice des prédictions,
2. Calculer le taux d'erreur pour chaque utilisateur suivant la formule :

$$\text{TauxErreur } i \square \square = \frac{\text{nbPredictionsFausses } i \square \square}{\text{nbRequetes } i \square \square} ,$$

où $\text{nbPredictionsFausses}(i)$ désigne le nombre de prédictions erronées pour l'utilisateur i et $\text{nbRequetes}(i)$ correspond au nombre de requêtes passées par l'utilisateur i (c'est-à-dire les deux tiers du nombre de requêtes définis lors de la création des bases de données utilisateurs),

3. Calculer le taux d'erreur moyen pour n exécutions du programme et par utilisateur :

$$\text{ErreurMoyenne } i \square \square = \frac{\text{TauxErreur } i \square \square}{n} ,$$

4. Calculer le taux d'erreur moyen : $ErreurMoyenne = \frac{ErreurMoyenne \bar{i}}{nbUtilisateurs}$
 où $nbUtilisateurs$ correspond au nombre d'utilisateurs définis lors de la création des bases de données utilisateurs.

Résultats et analyse

Une série de tests a été réalisée sur les bases de données utilisateurs de référence et de prédictions générées par la méthode présentée précédemment. Nous allons, dans cette partie, comparer les deux algorithmes de filtrage collaboratif : le coefficient de corrélation de Pearson et la similarité des vecteurs. Nous avons utilisé, pour cela, deux tailles de matrices (10 requêtes et 100 requêtes) et pour chaque matrice, nous avons fait varier le nombre d'utilisateurs, pour observer le comportement des algorithmes.

Par ailleurs, pour les algorithmes de Pearson et de similarité des vecteurs, une corrélation de 95% a été utilisée. Une corrélation aussi forte permet de ne considérer que les utilisateurs ayant réellement un profil similaire à l'utilisateur courant.

Enfin, cinq exécutions de chaque algorithme a permis d'obtenir les résultats que nous allons présenter dans la suite.

Tests avec 10 requêtes dans la base de données utilisateurs

Dans cette série de tests, les bases de données utilisateurs comprenaient un nombre de dix requêtes et nous avons fait varier le nombre d'utilisateurs. La figure 20 montre les erreurs de prédiction moyenne en fonction du nombre d'utilisateurs, pour les deux algorithmes.

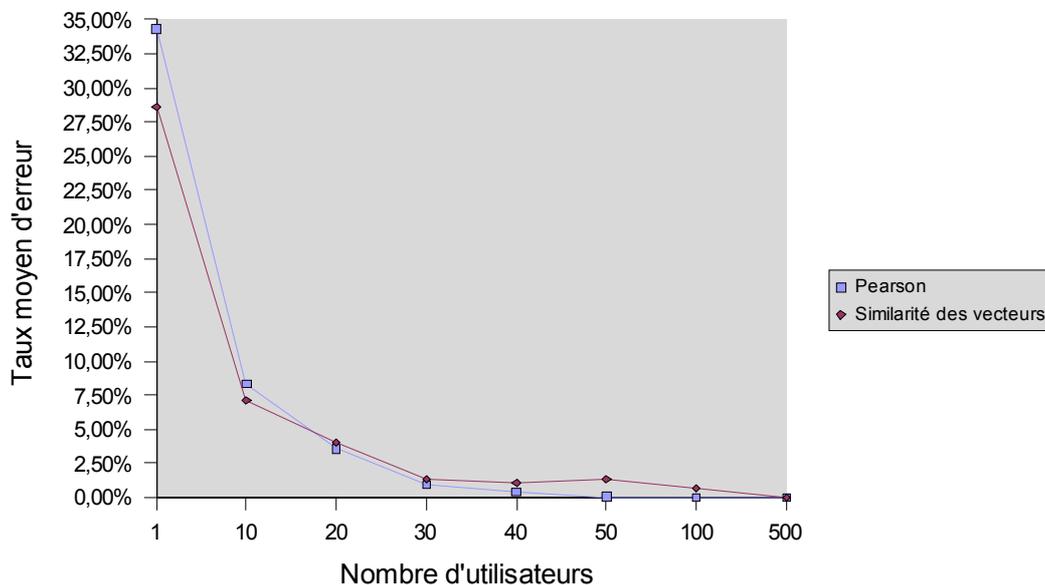


Figure 24 : Comparaison des méthodes de Pearson et de similarité des vecteurs sur 10 requêtes différentes

Tests avec 100 requêtes dans la base de données utilisateurs

Ici, le nombre de requêtes a été fixé à 100, et comme précédemment, nous avons fait varier le nombre d'utilisateurs. La figure 21 montre les erreurs de prédiction moyennes obtenues pour les deux algorithmes.

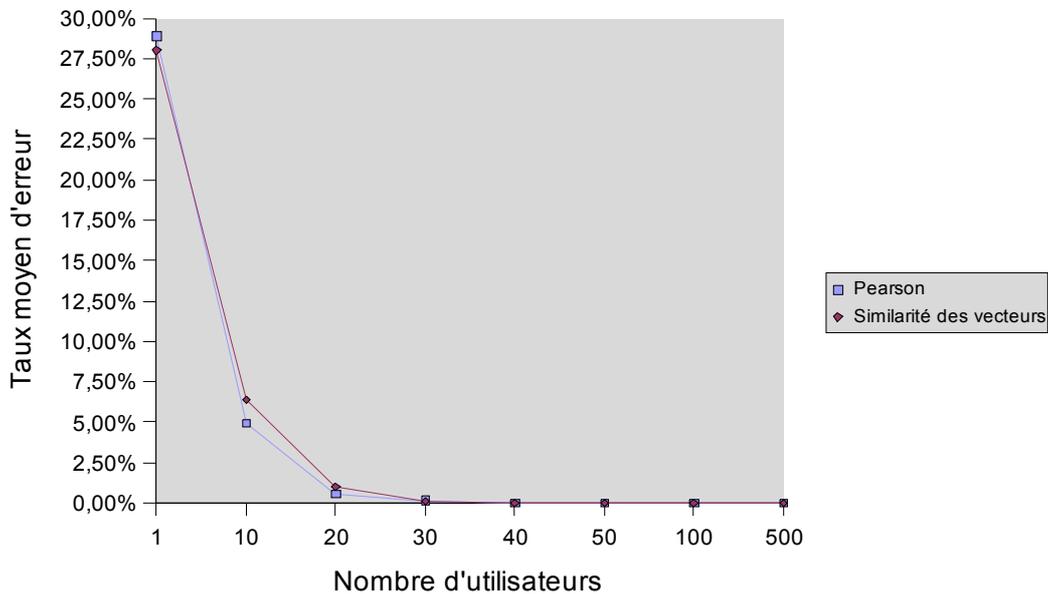


Figure 251 : Comparaison des méthodes de Pearson et de similarité des vecteurs sur 100 requêtes différentes

Analyse

Comme cela était prévisible, le nombre d'utilisateurs influe sur les prédictions. Lorsqu'il n'y a qu'un utilisateur dans la base de données utilisateurs, il n'est pas possible de lui trouver des utilisateurs similaires. Mais l'erreur de prédiction n'atteint pas pour autant 100%. Ceci est dû au fait que dans ce cas, le système prédit systématiquement le premier itinéraire. Et si l'utilisateur a choisi, dans la base de données de référence, l'itinéraire 1, la prédiction est considérée comme bonne. Il s'agit là d'un cas particulier de bonnes prédictions sans que l'avis d'autres utilisateurs intervienne. Ensuite, plus le nombre d'utilisateurs augmente et plus le taux d'erreur diminue, car le nombre d'utilisateurs isolés diminue. Les prédictions deviennent alors plus fiables et donc les algorithmes plus performants.

Que la base de données comprenne 10 ou 100 requêtes, les résultats sont quasiment identiques. La différence entre les deux méthodes est légèrement moins importante avec un grand nombre de requêtes. Sur un faible nombre de requêtes, le coefficient de corrélation de Pearson s'avère être une meilleure méthode pour effectuer les regroupements d'utilisateurs. En effet, avec le coefficient de corrélation de Pearson, le taux moyen d'erreur de prédiction tend plus rapidement vers 0 qu'avec la similarité des vecteurs. Ceci est un net avantage car le système peut fournir plus rapidement à un utilisateur la solution qui correspond réellement à ses besoins.

3.4 Evaluation de performance

Nous avons aussi effectué des tests *d'évaluation de la performance* du service de recherche d'itinéraire personnalisé. Nous avons mesuré le temps de réponse moyen du service par rapport au nombre d'utilisateurs contenus dans la base des utilisateurs.

La [figure 22](#) présente les temps de réponse suivant la méthode de filtrage utilisée au niveau des agents de gestion de profil. Les trois méthodes appartenant au modèle d'activité de l'agent de gestion de profil ont été testées, à savoir : une méthode cognitive (filtrage basé sur le profil de l'utilisateur : dernier choix) et deux méthodes sociales (vote majoritaire, Bayes).

Les résultats obtenus montrent que lorsqu'il s'agit d'un vote majoritaire, les temps de réponse augmentent en fonction du nombre d'utilisateurs enregistrés dans le système mais ces temps restent acceptables (moins d'une seconde pour 500 utilisateurs). Pour le filtrage à base de réseau bayésien (notre méthode de filtrage collaboratif basé sur les préférences et sur les comportements des utilisateurs), le temps de réponse est exponentiel à partir de 100 utilisateurs. Au delà de 100 utilisateurs, le temps de réponse dépasse 3 secondes. Nous pensons améliorer les performances en optimisant l'algorithme implémenté. Pour le filtrage basé sur le profil de l'utilisateur, le nombre d'utilisateurs enregistrés dans le système n'influe pas sur la performance du système.

D'autres évaluations de performance sont envisagées, par exemple, l'évaluation de la performance du service par rapport au nombre d'utilisateurs connectés simultanément au service ou l'étude de l'impact de la distribution physique des agents de PerSyst sur la performance globale du système d'information.

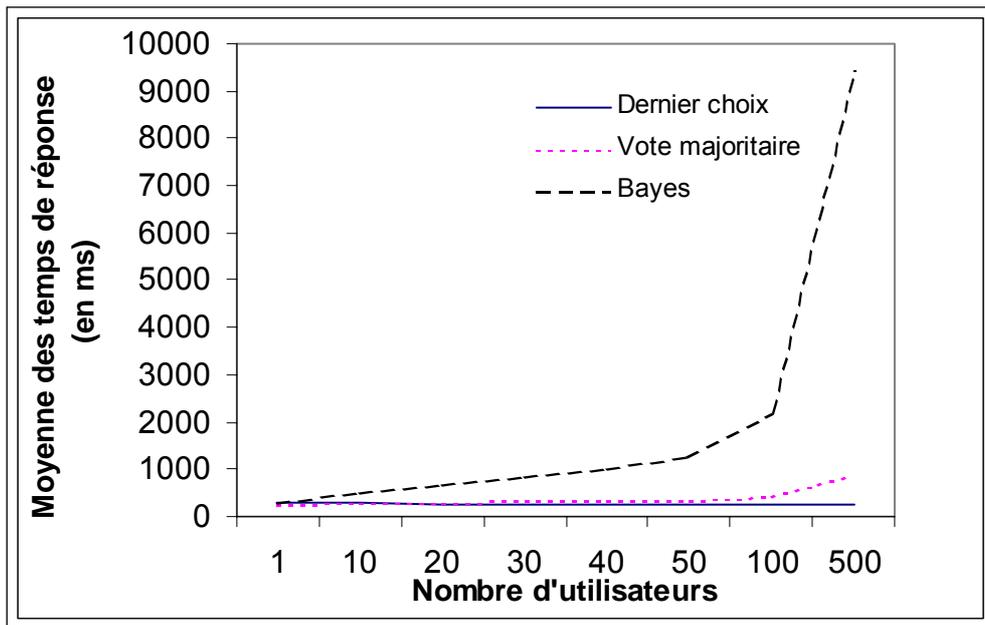


Figure 26. Temps de réponse du système en fonction du nombre d'utilisateurs enregistrés au service

4 Perspectives

D'autres services qui nécessiteraient des études ou des recherches beaucoup plus approfondies pourraient porter sur l'intégration des contextes, comme les conditions climatiques, l'état physique ou mental de l'utilisateur, dans la personnalisation de l'information transport. En effet, le choix d'un itinéraire par un utilisateur pourrait être motivé par rapport aux conditions climatiques, par exemple. Un utilisateur pourrait préférer le vélo lorsqu'il fait beau et choisir le bus lorsqu'il pleut ou lorsqu'il fait très froid. Un autre utilisateur pourrait aimer le métro mais favoriserait un taxi s'il est malade... La méthode de personnalisation que nous avons proposée pourrait être améliorée afin de tenir compte des contextes d'interaction. Toutefois, il n'est pas aisé de définir des contextes appropriés pouvant influencer la décision d'un utilisateur par rapport à un ensemble de choix. Et parfois, il est très difficile de fournir une méthode efficace pour capturer un contexte sans alourdir l'interaction homme-machine. Comment savoir que l'utilisateur est malade sans le lui demander ? Ce sont des problématiques intéressantes sur lesquelles il conviendrait de se pencher.

Nous commencerons par considérer les préférences des utilisateurs par rapport aux contextes relatifs aux conditions climatiques pour la recommandation des itinéraires. La température, l'humidité et la force du vent pourraient s'obtenir au travers de services oueb. A partir des informations (ville de départ, ville d'arrivée, date de départ, etc.) fournies par l'utilisateur lors de sa recherche d'itinéraire, il serait alors possible de déduire les conditions climatiques du déplacement. En appliquant notre méthode de personnalisation, il serait donc envisageable de recommander des itinéraires par rapport aux préférences des utilisateurs associées aux conditions climatiques. Nous pourrions continuer, ensuite, ces travaux pour la prise en compte d'autres contextes (luminosité, état physique, état psychique, etc.) aussi bien pour le filtrage du contenu que pour l'adaptation du contenant.

5 Conclusion

Le projet MOUVER.PERSO consiste à étudier et mettre en oeuvre des services d'accompagnement personnalisés autour de la mobilité quotidienne des étudiants. Nous avons centré ces services autour d'un agenda personnel capable de proposer une assistance à la planification liée à l'offre de transport. L'agenda proposé fonctionne par l'intermédiaire de pages web ainsi que via un PDA.

Son originalité réside en la proposition de solutions de transport correspondant aux rendez-vous planifiés.

A TERMINER...

6 Références

- [Anli 05a] Anli A., Grislin-Le Strugeon E., Abed M. (2005). *A Generic Personalization Tool based on a Multi-agent Architecture*. In C. Stephanidis (Ed.), Proceedings HCI International (Las Vegas, Nevada, July 22-27, 2005), Volume 7 - Universal Access in HCI: Exploring New Interaction Environments, Lawrence Erlbaum Associates, Mahwah, New Jersey, pp. 1-8, juillet.
- [Anli 05b] Anli A., Grislin-Le Strugeon E., Abed M. (2005). Une plate-forme de personnalisation basée sur une architecture multi-agents. *Revue des Nouvelles Technologies de l'Information (RNTI-E)*, 5, pp. 95-100.
- [Anli 05c] Anli A., Kolski C., Abed M. (2005). *Principes et architecture pour la personnalisation d'information en interaction homme-machine, application à l'information transport*. In Proceedings of IHM 2005, International Conference Proceedings Series, ACM Press, Toulouse, pp. 123-130, septembre.
- [Boidin 05] Boidin E. *Apprentissage automatique pour la personnalisation : Application aux transports terrestres de personnes*. Rapport de DEA, Université de Valenciennes et du Hainaut-Cambrésis, juillet 2005.
- [Marzloff 03a] Marzloff B. (2003). *Prospectives de mobilité urbaine. Du monomodal à l'intermodal, place de l'information et des canaux d'information dans les pratiques des modes de transport urbains*. Rapport de projet PREDIM-MOUVER. MEDIA MUNDI (Groupe CHRONOS), SEREHO (Groupe CHRONOS), ACT FIELD WORK, juin.
- [Marzloff 03b] Marzloff B., Rudeau L. (2003). *Prospectives de mobilités intermodales et multimodales. Enquête sur le bassin de mobilité de l'agglomération de Lille*. Rapport de projet PREDIM-MOUVER. MEDIA MUNDI (Groupe CHRONOS) et SEREHO (Groupe CHRONOS), mars.
- [NWG 05] Network Working Group. (2005). *Guideline for use of XML with iCalendar elements – draft-harexcalendar-03*, mai 2005.
- [Petit-Rozé 05a] Petit-Rozé C. *Personnalisation de l'information à base d'agents intelligents*. Thèse de l'université de Valenciennes et du Hainaut-Cambrésis, déc. 2003.
- [Petit-Rozé 04] Petit-Rozé C., Anli A., Grislin-Le Strugeon E., Abed M., Uster G., Kolski C. (2004). Système d'information transport personnalisée à base d'agents logiciels. *Génie Logiciel*, 70, pp. 29-38.
- [Petit-Rozé 05b] Petit-Rozé C., Grislin-Le Strugeon E. (2005). *Modélisation d'interactions par et avec des agents en personnalisation d'information*. In A. Herzig, Y. Lespérance et A.-I. Mouaddib (Eds.), Actes des Troisièmes Journées Francophones Modèles Formels de l'Interaction - MFI'2005 (Caen, France, 25-27 Mai 2005), Cepadues-Editions.
- [Rizcallah 00] Rizcallah M. (2000). *Construire un annuaire d'entreprise avec LDAP*. Eyrolles, Paris, 2000.
- [IETF] Standard LDAP version 3 disponible à l'adresse : <http://www.ietf.org/rfc/rfc2251.txt>

[12] Standard XSL disponible à l'adresse : <http://www.w3.org/Style/XSL>

7 Annexe

Un ensemble de fonctionnalités sont visibles à titre de démonstration sur le site *Mon-Service-Transport.com* à l'adresse : <http://83.145.86.88/masc/?INSTANCE=sitp&setlanguage=fr>

Un enregistrement vidéo d'environ 5 mn est en particulier consultable en suivant le lien *Demos* du menu de gauche dans la page d'accueil du site.

Remarque :

Pour des essais directs, il est nécessaire de s'inscrire (lien *Inscription* du menu de gauche dans la page d'accueil). Une fois l'inscription réalisée (remarque : l'envoi du courriel correspondant n'est pas encore opérationnel), il est possible de s'authentifier (identifiant et mot de passe à entrer en partie droite de la page d'accueil) afin d'accéder au portail personnalisé.

Le menu de gauche présente alors un ensemble de fonctionnalités mais s'agissant d'un prototype en cours de développement, toutes les fonctionnalités ne sont pas implémentées.