



HAL
open science

Évolution d'une application orientée services vers les Services Web

André Thépaut, Jérôme Kerdreux, Yvon Kermarrec, Christophe Lohr,
Philippe Normand

► **To cite this version:**

André Thépaut, Jérôme Kerdreux, Yvon Kermarrec, Christophe Lohr, Philippe Normand. Évolution d'une application orientée services vers les Services Web. *Génie logiciel: le magazine de l'ingénierie du logiciel et des systèmes*, 2006, 76, pp.35 - 41. hal-01759101

HAL Id: hal-01759101

<https://hal.science/hal-01759101>

Submitted on 28 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Évolution d'une application orientée services vers les *Web Services*

André Thépaut - Jérôme Kerdreux - Yvon Kermarrec - Cristophe Lohr - Philippe Normand

École Nationale Supérieure des Télécommunications de Bretagne
Technopôle Brest-Iroise –BP 832- 29285 Brest Cedex France

Tél : +330 229001437 Télécopie : +330 229001282 – Courriel : andre.thepaut@enst-bretagne.fr

Résumé :

La proportion de personnes âgées dans la population augmente, mais le nombre d'aidants ne suit pas cette évolution. Le projet T@PA « Télé-@ssistance pour Personnes Agées » se propose d'aider à rompre l'isolement des personnes âgées, qui souhaitent rester à domicile, en fournissant un environnement simple d'utilisation ainsi que des applications d'accès à la Société de l'Information.

Le cœur de la plate-forme est constitué d'un téléviseur et d'un ordinateur domestique qui est connecté au «réseau social » de la personne âgée. La plate-forme offre un ensemble de services, accessibles via une télécommande simplifiée, et qui favorisent une proximité virtuelle avec l'entourage.

Dans cet article nous présentons l'architecture de la plate-forme T@PA qui est testée à Brest par quelques seniors volontaires et leurs proches depuis la mi-novembre 2004. Nous présentons également les premières conclusions portant sur les choix d'implémentation technologiques effectués dans le cadre du projet.

Le fonctionnement de T@PA repose sur l'utilisation de services communiquant entre eux via le protocole XML-RPC, mais la technologie actuelle évoluant très vite, nous avons souhaité la transition vers une architecture à base de Web Services (WSDL, UDDI).

L'objectif est de comprendre comment faire évoluer le logiciel T@PA, qui n'a pas été conçu à la base pour ce type de technologie, et de comparer les différentes technologies actuelles en terme d'efficacité et d'évolutivité.

Mots clés : Web Services, XML-RPC, SOAP, WSDL, UDDI, client léger, client semi-léger.

1. Introduction

Le projet T@PA (« Télé-@ssistance pour Personnes Agées ») associe une équipe de chercheurs composée d'informaticiens, de sociologues [1] et d'économistes à des acteurs des services sociaux impliqués dans l'accompagnement des personnes âgées demeurant à leur domicile. Le service T@PA [2] peut être utilisé par la personne âgée et/ou son entourage. Il a fallu prendre en compte les attentes des personnes vieillissantes et favoriser les relations avec les réseaux affectifs, sociaux ou professionnels qui les entourent.

L'équipement présent au sein du domicile de la personne âgée est constitué du téléviseur et d'un ordinateur domestique (machine compacte et silencieuse) qui est connectée au « réseau social » par un accès Internet. La plate-forme offre un ensemble de services, accessibles via une télécommande simplifiée, garantissant une proximité virtuelle de l'entourage.

Deux grandes familles de services sont actuellement proposées par la plate-forme T@PA :

- des services d'aide à la relation avec le réseau communautaire (couplage avec les services Web municipaux, diffusion d'informations sur le quartier, intégration de films proposés par l'association TV Résidences à Brest, etc.)
- des services d'aide à la relation avec le réseau affectif (photos, vidéos, messages diffusés par la famille, les amis, etc.) qui permettent à la personne âgée de maintenir le contact avec les personnes de son choix.

La suite de l'article présente la plate-forme T@PA, les premiers résultats de l'expérimentation auprès d'une vingtaine d'utilisateurs (personnes âgées, famille, réseau professionnel) et une discussion autour de l'apport d'une transition vers les Web Services.

2. Architecture de T@PA

L'architecture de la plate-forme T@PA est articulée autour d'un serveur et d'équipements mis à disposition des personnes âgées et de leur entourage. Le serveur héberge les services offerts par la plate-forme [3] et les rend accessibles via plusieurs types d'interfaces. Le premier type d'interface est destiné aux personnes âgées. C'est une interface dédiée, visualisée sur le téléviseur de la personne âgée. Le boîtier T@PA est connecté au réseau haut débit via un ordinateur domestique, et contrôlable via une télé-commande simplifiée. Le second type d'interface est destiné aux personnes de l'entourage qui disposent de trois alternatives pour transmettre des informations :

- un site internet,
- la messagerie électronique classique
- l'envoi de message multimédias par téléphone portable.

La figure 1 présente une vue générale de l'architecture de la plate-forme et des relations entre les différents types de participants.

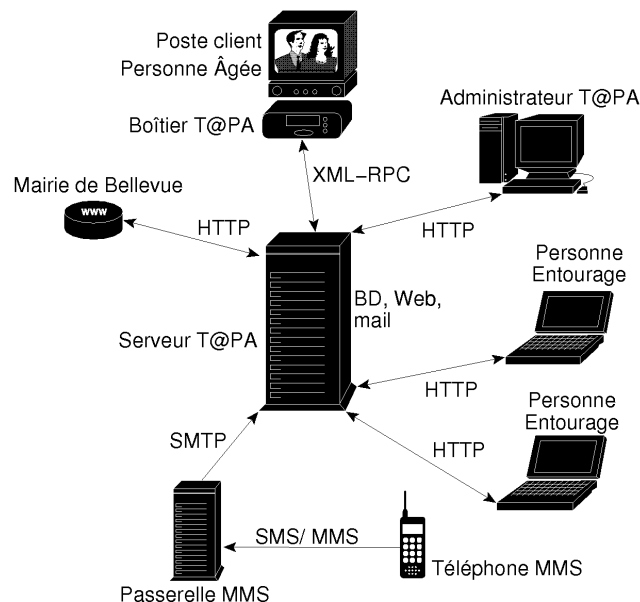


Figure 1 : Architecture générale de T@PA

Les sections suivantes présentent les entités et les protocoles mis en oeuvre au sein de la plate-forme.

2-1 Le(s) serveur(s) T@PA

La plate-forme T@PA repose sur l'utilisation d'un serveur hébergé à l'ENST Bretagne. Ce serveur qui sert de lien entre les différents acteurs du projet (personnes âgées, personnes de l'entourage, administrateur système, services communautaires, etc.) héberge actuellement trois catégories de modules logiciels (Figure 2). Les interfaces publiques permettent l'accès aux services utilisateur via une interface Web et/ou via une interface d'accès MMS et ne suivent pas l'architecture d'un service T@PA.

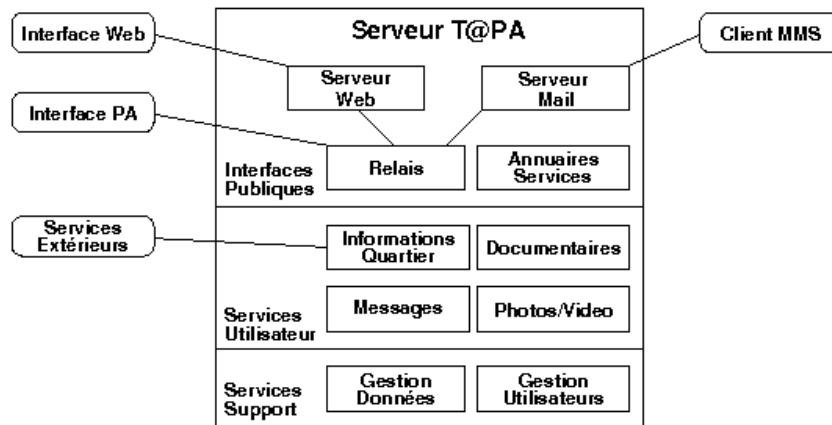


Figure 2 : Relations client/service

Les services *support* et les services *utilisateur* ont été conçus de manière à permettre leur distribution (ils sont accessibles à distance via une interface XML-RPC [4]) et à séparer clairement les aspects non fonctionnels des aspects fonctionnels. En particulier, un service T@PA est constitué principalement d'une interface d'utilisation (ensemble de fonctions rendues disponibles à d'autres services), d'une interface graphique spécifique pour le poste de la personne âgée, d'une interface Web et de l'ensemble de services requis pour son exécution (les dépendances). La figure ci-dessous présente la structure élémentaire d'un service. Pour créer un service le développeur hérite de la classe `Service`, en redéfinit des parties et ajoute des méthodes qui constituent l'API du service.

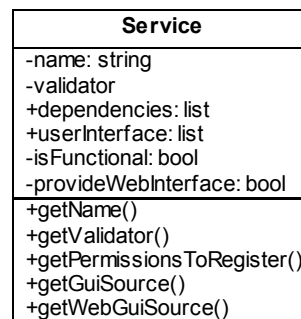


Figure 3 : La classe `Service`

Afin d'implémenter les services requis, notre choix s'est porté sur la plate-forme logicielle OSE [5]. Cette plate-forme inclut un annuaire de services et gère l'éventuelle distribution de ceux-ci sur plusieurs serveurs. Chaque service se matérialise par un processus exécuté sur la machine. Du point de vue du développeur, un service se présente sous la forme d'une classe Python, dont certaines méthodes sont exportées via OSE, et disponibles soit publiquement, soit aux utilisateurs T@PA authentifiés. La plupart des services constituent une interface à la base de données qui regroupe les informations sur les utilisateurs, certaines données liées aux services, couplées à un stockage direct sur disque dur pour les données multimédias (photos, vidéos).

T@PA permet également d'utiliser la technologie MMS (*Multimedia Message System*) des téléphones portables

de dernière génération. De cette manière les personnes de l'entourage peuvent utiliser leur portable pour communiquer avec la personne âgée en lui transmettant des messages textuels, des photos, ou de petites vidéos. Les MMS ainsi composés transitent par l'opérateur de télécommunication et parviennent sous forme de courriel à la plate-forme T@PA. Les données utiles de ces messages sont traitées et déléguées aux services T@PA, par exemple le service de *gestion d'albums multimédias* ou le service de *messagerie*.

2-2 Le poste client PA (Personne Âgée)

Au domicile de la personne âgée, un mini-pc domestique (Figure 4) est relié au téléviseur. Cet ordinateur se connecte à l'Internet haut débit via un modem-routeur ASFI (Accès Sans Fil à Internet : *WiFi*) pour accéder aux services T@PA. Le choix d'une technologie sans fil et d'un mini boîtier silencieux a été motivé par le désir des personnes âgées de limiter l'intrusion technologique dans leur cadre de vie.

Pour accéder aux services proposés par T@PA, nous avons mis en place une interface (Figure 6) qui repose sur une succession de menus simples. Ces menus sont accessibles via une télécommande simplifiée (Figure 5), solution qui a été privilégiée par les personnes âgées par rapport à des solutions plus élaborées (écran tactile, souris, commande vocale, etc.). Cette télécommande, associée à un petit mémo d'utilisation, semble bien acceptée par les personnes. Un récepteur infra-rouge, placé à proximité du téléviseur, capte les trames de commandes et les transmet à l'ordinateur domestique via une ligne série.



Figure 4 : Ordinateur domestique



Figure 6 : Interface TV



Figure 5 :
Télécommande

Le logiciel client est de type semi-léger. Il comporte une partie statique, installée sur le disque dur de l'ordinateur, dont le rôle est de télécharger le reste de l'interface graphique à partir du serveur. Cette solution présente des avantages en termes de maintenabilité du code et de déploiement. L'interface graphique utilise la bibliothèque de composants graphiques Gtk+, associée à un thème d'affichage adéquat pour le téléviseur (polices de grande taille, couleurs, ...). A l'initialisation du client, une session utilisateur est ouverte. Cette session contient une pile d'événements reçus par le serveur. Ces événements permettent à l'interface de réagir en temps réel à l'arrivée de nouveau contenu (message, photos, ...) et de se mettre à jour (sans redémarrage de l'ordinateur domestique) en fonction de l'activation/désactivation de services sur le serveur.

2-3 Le poste client PE (Personne de l'Entourage)

Il n'y a pas de dispositif matériel ni de logiciel spécifique installé chez les personnes de l'entourage. Pour utiliser les services fournis par la plate-forme T@PA les personnes de l'entourage se connectent, par l'intermédiaire de leur navigateur, au site Web du projet (<http://tapa.enstb.org>). Après une phase d'authentification, elles peuvent accéder, via l'interface et des liens hypertextes, aux différents services disponibles.

Le site Internet agit en tant que client aux services T@PA et utilise (de même que le client PA) le protocole XML-RPC pour les communications avec les services. De cette manière il est possible d'héberger le site Internet sur un serveur différent de celui hébergeant les services.

Les membres de l'entourage de la personne âgée peuvent également utiliser d'autres moyens tiers pour accéder à la plate-forme T@PA. Par exemple l'expédition de mini-messages et/ou photos via le téléphone portable suit les étapes ci-après:

Génie Logiciel

La PE (Personne de l'Entourage) compose un MMS sur son portable à destination de la personne âgée qui dispose d'une adresse de courrier électronique,

- le MMS transite par l'opérateur de télécommunications vers le serveur mail du serveur T@PA,
- le courriel est transmis à un mini-client T@PA qui diffuse les données utiles (texte, photo, vidéo) au sein de la plate-forme,
- la personne âgée est avertie de l'arrivée d'un nouveau media et guidée via l'interface graphique vers ce dernier.

Puisqu'un MMS est assimilable à un courrier électronique, la personne de l'entourage peut également utiliser sa messagerie électronique habituelle pour contacter la personne âgée.

3. Retour d'expérience

Le projet a été expérimenté au sein de 4 familles brestoises, totalisant ainsi une vingtaine d'utilisateurs (famille, mairie de quartier, etc.). Des études sociologiques et économiques sont en cours de réalisation et ont produit des premiers résultats [1]. Sur le plan technique nous avons pu également retirer des enseignements de cette première expérimentation.

3-1 Modularité de la plate-forme

L'architecture logicielle de la plate-forme T@PA s'appuie sur un ensemble de services. Le fait que la plate-forme soit modulaire en garantit sa pérennité car il est aisé d'ajouter dynamiquement de nouveaux services qui sont *de facto* disponibles pour les utilisateurs. De plus, la plate-forme OSE propose la distribution des services, ce qui dans le contexte d'une mise à l'échelle de l'application en cas de forte utilisation de T@PA est un atout non négligeable.

3-2 Atouts et inconvénient de la légèreté de XML-RPC

La technologie XML-RPC étant simple et légère, les communications de type client/service sont rapides et le client est réactif, mais cela nécessite une attention particulière quant à la quantité d'informations nécessaire. En particulier l'interface client destinée aux personnes âgées requiert un niveau élevé de disponibilité. Il faut donc prêter une attention particulière aux volumes de données que ce client traite.

Les communications inter-services se faisant également via XML-RPC, il faut éviter de multiplier les dépendances entre les services car cela peut ralentir les performances de la plate-forme de façon significative et empêcher le passage à l'échelle. Par exemple, chaque appel de méthode aux services par un client authentifié provoque un appel au service de gestion des utilisateurs afin de vérifier que la session ouverte par l'utilisateur n'a pas expiré. Il faut donc veiller aux bonnes performances de cette dernière routine.

L'absence de langage de description au sein de la technologie XML-RPC, nécessite une bonne connaissance de la plate-forme lors du développement et de ce fait ne garantit pas une interopérabilité idéale des services. Par exemple, si un développeur désire créer un nouveau service, il doit posséder une bonne connaissance du service de gestion des utilisateurs. Nous poursuivons actuellement des études autour des technologies telles que WSDL [6], SOAP [7] et WSFL [8] dans le but d'apporter au projet une meilleure interopérabilité.

4. Passage aux Web Services

L'échange d'informations structurées et typées entre pairs dans un environnement décentralisé et distribué est une préoccupation majeure de l'informatique. De nombreuses technologies ont été développées dans ce but (CORBA, RMI...) et de nombreuses plate-formes logicielles implémentent ces notions d'objets distribués. Les Web Services apportent une autre approche pour la description de services avec références précises au comportement de ceux-ci (via WSDL et WSFL), à la description des propriétés statiques grâce notamment à DAML-S [9] et aux ontologies. Ces différents niveaux de description permettent d'indexer précisément les services et pour l'utilisateur de les découvrir dynamiquement et d'interagir avec eux.

Les Web Services tendent à apporter des solutions pour le développement des applications en mettant en avant la notion de service rendu accessible et ouvert. Aujourd'hui, les grands acteurs de l'industrie œuvrent à développer

des standards (WSDL, SOAP, UDDI...) et des outils pour ces Web Services. Dans le cas du projet T@PA, nous avons souhaité évaluer l'apport d'une transition vers les Web Services en repensant l'architecture globale du projet, de manière à apporter une ouverture et une interopérabilité renforcée tout en assurant la compatibilité avec les standards en cours. Cependant les Web Services sont encore en évolution et leur utilisation nécessite de prendre des précautions dans le cadre d'un projet opérationnel. Nous avons donc décidé de mener une évaluation de ces technologies pour savoir si elles pourraient devenir le socle du projet T@PA. Avant d'envisager une nouvelle version de T@PA, il nous faut d'abord vérifier que cette transition n'affecte pas les performances de la plate-forme et prenne en charge toutes les conditions de sécurité et d'évolutivité qui ont été définies lors de la conception. Une difficulté de cette étude de faisabilité est que T@PA n'a pas été spécifiquement conçu ni développé pour le partage de services ou l'ouverture avec d'autres applications.

4.1 Utilisation de WSDL et SOAP

WSDL permet de décrire un service, et ses interfaces, afin qu'il puisse être invoqué depuis l'extérieur sans connaître les détails d'implémentation. Grâce à cette description, le client peut accéder à toutes les informations pour pouvoir utiliser le service (URL du serveur, description des fonctions disponibles, etc.) ; cette description est ensuite utilisée pour générer les objets et appels nécessaires pour la mise en relation du client et du serveur.

De son côté, SOAP a pour but de fournir un protocole d'échange efficace entre client et serveur s'appuyant sur le protocole HTTP et le standard XML. L'intérêt principal de SOAP est de pouvoir faire interagir des composants distants, indépendants et hétérogènes. Il nous donne la possibilité de passer à une conception orientée objet, simplifiant ainsi le travail du développeur.

La conception ou l'intégration de nouveaux services est simplifiée grâce à ces deux technologies. Un développeur peut, par exemple, développer son propre service dans le langage de son choix, puis fournir une description WSDL aux autres membres du projet, description dans laquelle seront décrites toutes les informations nécessaires aux applications clientes.

Cependant le passage vers ces deux technologies ne peut se faire sans conséquences car la taille des données échangées augmente de manière importante, proportionnellement à la complexité des types utilisés. À cela s'ajoute d'autres problèmes moins conséquents tels que l'obligation pour le client d'analyser et de valider le fichier WSDL avant de l'utiliser. Au final le temps de réponse peut devenir prohibitif dans le cas d'une application réactive. L'un des objectifs de cette étude est de vérifier que l'accroissement de la taille des flux ne devient pas nuisible au bon fonctionnement de la plate-forme.

4.2 Une interdépendance contraignante.

Afin de sécuriser la plate-forme de manière efficace, un système de contrat [10] a été mis en place dans le projet T@PA. À tout moment, un utilisateur désireux d'invoquer une fonctionnalité d'un service passera par une vérification de ses droits via le « Service Utilisateur », sorte de centre de gestion des utilisateurs de T@PA. Cette méthode permet d'assurer en permanence une sécurité optimale mais multiplie les échanges d'informations, et ralentit donc le temps de réponse d'une requête. Dans le cadre de notre plate-forme avec XML-RPC, nous avons mis en évidence le nombre important de ces interactions. Il nous faut donc vérifier dans quelle mesure le passage à une technologie SOAP/WSDL affecte le temps de réponse, sachant que le service utilisateur devra recevoir et renvoyer des messages encore plus volumineux.

De plus une inter-dépendance se crée entre les services, par exemple le service « Messagerie » a besoin du « Service Utilisateur » pour fonctionner, ceci apporte une contrainte pour le développeur qui ne serait contournable qu'en modifiant la manière dont sont gérés les utilisateurs.

4.3 Evaluations.

Durant la première partie de ce travail, les différents services ont été intégralement décrits via des fichiers WSDL puis les premiers comparatifs ont pu être faits entre les messages SOAP et les messages XML-RPC. Comme nous pouvions le supposer, la taille des données transmises et le temps de réponse augmentent effectivement dans le cas de l'architecture à base de WebServices. Cette augmentation est bien plus conséquente dès que les types des objets transmis sont complexes. Notre objectif est de récolter un maximum de données (benchmark) pour comparer au mieux les deux technologies XML-RPC et SOAP afin de savoir dans quel cas il

semble plus judicieux d'utiliser l'une ou l'autre.

Cette identification nous permettra également de localiser les parties du logiciel qui présentent des risques importants d'engorgement et de modifier en conséquence l'architecture globale de notre plate-forme.

Protocole de tests

Deux plate-formes T@PA sont mises en place pour les tests, sur la même machine. Deux bases de données jumelles sont utilisées. La seule différence entre les deux plateformes est l'intergiciel utilisé. Notre objectif est de comparer la technologie XML-RPC à la combinaison de SOAP et WSDL.

Les tests ayant lieu sur la même machine, la latence réseau n'est pas prise en compte. Les fonctionnalités de service testées ont la même API.

Test sur le nombre d'utilisateurs de la plate-forme

On se propose d'ajouter de nouveaux utilisateurs et de récupérer la liste des utilisateurs au fur et à mesure suivant l'algorithme suivant :

```
for i in 1..20 do
  addUsers(5)
  timeXMLRPC("getUsers")
  timeSOAP("getUsers")
end for
```

En implémentant cet algorithme et en collectant les données portant sur les temps d'exécution par rapport au nombre d'utilisateurs enregistrés au sein de la plate-forme, on obtient le graphe représenté sur la figure 7.

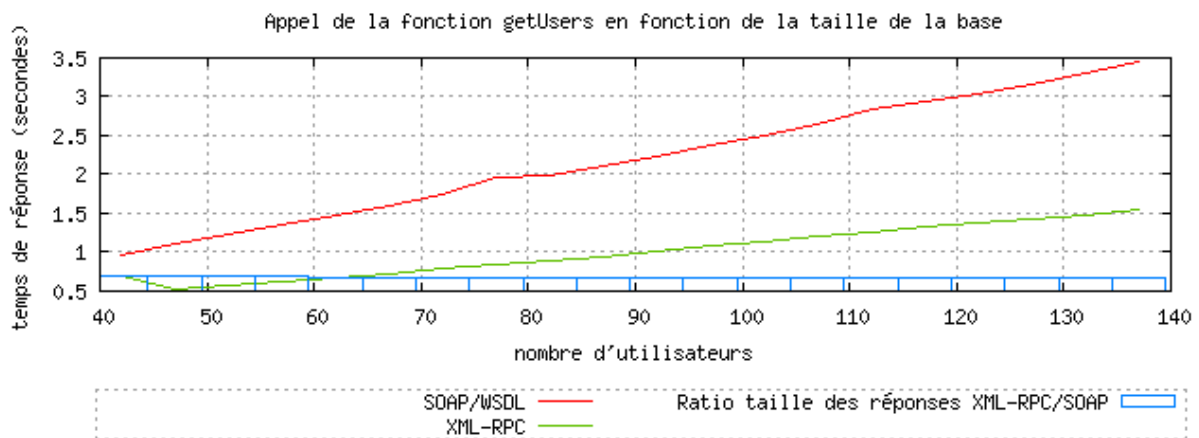


Figure 7 : Evolution de la réactivité de la plate-forme en fonction du volume de données transmises

Le service de gestion des utilisateurs n'ayant pas de dépendance à d'autres services T@PA, l'appel de la méthode « getUsers » ne provoque aucun autre appel RPC du côté serveur. Le test suivant est réalisé sur un service ayant des dépendances vers d'autres services T@PA.

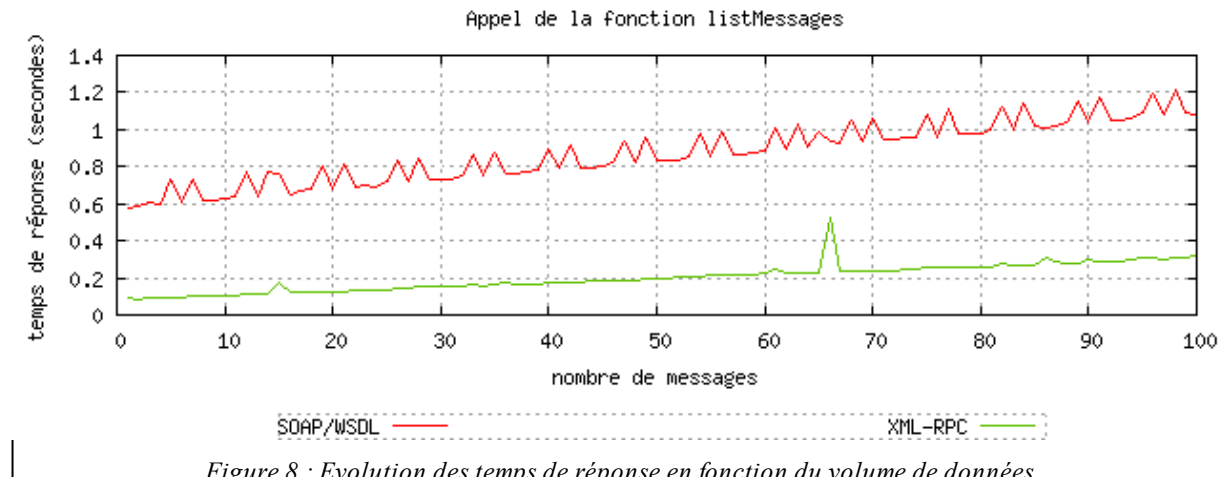
Test sur l'influence du système de contrat en SOAP

- A chaque méthode de service il est possible d'associer deux types de contrat : un contrat de type *pre* qui vérifie la validité des paramètres passés à la méthode avant l'exécution de celle-ci,
- un contrat de type *post* qui vérifie les contraintes portant sur le résultat de la méthode.

Le test suivant porte sur la méthode « listMessages » du service de messagerie T@PA. À cette méthode est associé un contrat de type *post* qui garantit que l'utilisateur ne pourra pas lire les messages émanant de personnes

qui ne figurent pas dans le réseau social de celui-ci.

Le scénario du test est le suivant : deux sessions utilisateur sont ouvertes ; le premier utilisateur envoie un message au second qui ensuite récupère la liste des messages qu'il a reçus. On itère le scénario, la liste des messages augmente en taille puisque les messages ne sont pas supprimés.

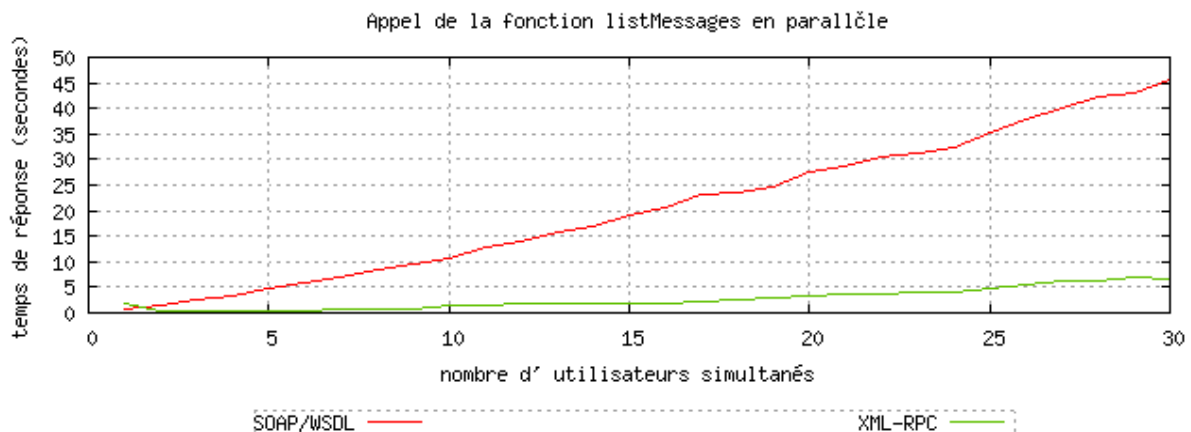


On constate une légère saturation de la plate-forme basée sur XML-RPC quand le nombre de messages atteint 66, mais globalement l'évolution du temps de réponse est constante. Par contre la courbe de la plate-forme basée sur SOAP et WSDL présente une évolution singulière que nous n'avons pas réussi à expliquer. Globalement le facteur « réactivité » entre les deux plate-formes T@PA est de l'ordre de 2.5, en faveur de la solution basée sur XML-RPC.

Test sur l'influence du nombre d'utilisateurs simultanés

Le projet T@PA se doit de respecter une certaine qualité de service, notamment en ce qui concerne les temps de réponse aux requêtes des applications clients. Il faut en effet être assuré que la navigation reste le plus fluide possible pour ne pas nuire à l'utilisateur. Il peut être intéressant de connaître comment évoluerait cette qualité de service en fonction du nombre d'utilisateurs de T@PA. En effet, dans le cas d'une utilisation à grande échelle de T@PA, il semble évident que des pics d'affluence apparaîtront en fonction des moments de la journée, entraînant des risques de saturation du serveur.

Les tests doivent s'intéresser ici à une situation bien particulière : comment le temps de réponse va-t-il évoluer lors de requêtes simultanées de nombreux utilisateurs ? Le test suivant mesure les performances de la plate-forme dans le cadre d'une simulation d'utilisateurs qui consultent un service unique au même instant.



La figure 9 met clairement en évidence que la plate-forme basée sur SOAP et WSDL est vite saturée dès que le nombre d'utilisateurs connectés augmente. Il est en effet difficilement acceptable pour un utilisateur de patienter 45 secondes pour lire ses messages en période de forte affluence (i.e. 30 utilisateurs simultanés) sur la plate-forme T@PA. Les performances de la plate-forme T@PA basée sur la technologie XML-RPC restent acceptables, même en période de forte affluence.

5. Décomposer et s'ouvrir vers l'extérieur grâce à UDDI

Les grands acteurs de l'industrie informatique se sont associés afin de produire des outils facilitant les partages de moyens entre les entreprises ou organisations proposant des services. Pour faire cela il fallait trouver des moyens pour pouvoir classer les services en catégories distinctes et proposer ces services au monde extérieur, c'est ainsi qu'est apparue la notion d'annuaire UDDI [11]. Les services, une fois répertoriés dans cet annuaire peuvent ensuite être retrouvés et proposés grâce à leur description. Les techniques pour répertorier, classer et retrouver les services sont encore en pleine évolution [12] mais permettent déjà une description précise des services.

Les annuaires de type UDDI permettent d'indexer et de retrouver les services à partir de leurs propriétés statiques. L'annuaire universel proposé par UDDI apparaît comme irréaliste du fait de son universalité et nous proposons plutôt un annuaire local utilisable par la communauté des utilisateurs de T@PA. Plusieurs services proposés par T@PA peuvent être réutilisés dans d'autres applications ; par exemple T@PA propose des services tels que les informations de la ville ou du quartier, ce service peut être réutilisé par d'autres logiciels. Une utilisation d'un annuaire UDDI public ou privé serait alors un gage d'ouverture vers l'extérieur et permettrait une réutilisation des services conçus.

L'annuaire de services de T@PA peut être remplacé par un annuaire local de type UDDI permettant à la plate-forme de proposer une meilleure accessibilité à ses services. Ce remplacement permettrait au client d'étendre les critères de recherche et de retrouver plus facilement les services dont il a besoin [9].

Cependant ce passage nous paraît relativement complexe dans l'immédiat du fait des dépendances utilisées au sein des services. Par exemple si un client désire récupérer les informations de son quartier, il devra s'identifier sur le service utilisateur pour recevoir un numéro de session. Grâce à ce numéro de session, il peut appeler la fonction pour récupérer les informations voulues en utilisant le service « InfosVille », mais là encore une vérification des droits de l'utilisateur sera effectuée grâce au numéro de session donné.

Nous avons donc un choix à faire : rendre les services indépendants ce qui implique des modifications de T@PA, ou bien trouver un moyen de décrire ces dépendances. Cette description permettra à une personne extérieure de comprendre comment les services doivent être utilisés. Il est possible de décrire plus finement les séquences d'utilisation des services grâce à WSFL, qui est un langage basé XML, conçu par IBM, permettant de décrire l'orchestration d'un ensemble de services dans le but d'atteindre un objectif précis.

6. Conclusion

La plate-forme T@PA constitue un exemple réel d'application orientée WebServices. Le projet a été expérimenté avec succès auprès d'une vingtaine d'utilisateurs (personnes âgées, entourage, personnel de la ville de Brest). Du point de vue technique il est envisagé d'utiliser les technologies SOAP et WSDL dans le but de déployer la plate-forme à plus grande échelle. Toutefois, les performances de la plate-forme actuelle, utilisant la technologie XML-RPC, sont acceptables dans la mesure où les développeurs de Services T@PA sont conscients des volumes d'informations transitant entre les services et le client.

Les standards et technologies que nous avons retenus pour l'évolution de la plate-forme T@PA apportent une ouverture non négligeable du fait de la compatibilité avec les travaux relatifs au *web sémantique*.

Nous avons exposé les objectifs de cette étude et les obstacles à franchir. Ces travaux n'ont pas pour but de démontrer que telle technologie est meilleure qu'une autre, mais plutôt de comprendre quel est le bon choix à faire dans le cas particulier du projet T@PA. D'autres aspects sont à prendre en compte tels que la facilité de développement, d'intégration et de maintenance de services (internes ou externes) dans le contexte d'une diffusion à plus grande échelle de T@PA.

Le changement de technologie middleware au sein de la plate-forme T@PA a clairement mis en évidence un

problème de conception de l'architecture logicielle. En effet chaque appel de méthode des services T@PA induit une vérification de session du client au niveau du service de gestion des utilisateurs. Si la communauté des utilisateurs de T@PA s'étend, cela peut poser des problèmes importants. On se propose donc d'implémenter un système de passeport (probablement basé sur Kerberos[13]). Quand un utilisateur désire utiliser un service, un passeport est généré pour ce service. Le passeport est chiffré avec la clef du service, ce dernier est seul capable de déchiffrer le passeport qui peut contenir certaines informations personnelles et les privilèges de l'utilisateur. Ainsi, il n'est plus nécessaire de solliciter le service de gestion des utilisateurs à chaque appel de méthode.

7. Remerciements

Le projet T@PA est soutenu par la Fondation d'Entreprise SFR/Cegetel, la Communauté Urbaine de Brest et le Conseil Général du Finistère. Les auteurs tiennent également à remercier les personnes volontaires ayant participé à l'expérimentation.

8. Bibliographie

- [1] Hélène Trellu « *Les personnes vieillissantes et la télé-assistance : privilégier la dimension relationnelle* », Colloque « Les nouvelles technologies dans la cité », 9 décembre 2004, Rennes.
- [2] A. Thépaut, J. Kerdreux, P. Normand, M.T. Segarra « *De la télé-assistance pour personnes âgées à la Télé-relation : évolution du projet T@PA* », Colloque « Les nouvelles technologies dans la cité », 9 décembre 2004, Rennes.
- [3] P. Normand, Rapport de stage DESS Architecture et Systèmes d'Information en Réseau (ASIR), Université de Bretagne Sud (UBS, Vannes), 2004, ENST-Bretagne.
- [4] XML-RPC : Specification version 3, Updated 6/30/03 DW, <http://www.xmlrpc.com/spec/>
- [5] OSE: Overseas Telecommunications Corporation (OTC) Software Environment, <http://ose.sourceforge.net/>
- [6] WSDL: Web Services Description Language version 1.1 03/15/01, <http://www.w3.org/TR/wsdl>
- [7] SOAP: Simple Object Access Protocol version 1.2 06/24/03, <http://www.w3.org/TR/soap12-part1/>
- [8] WSFL: Web Services Flow Language version 1.0 Prof. Dr. Frank Leymann (Distinguished Engineer; Member IBM Academy of Technology, IBM Software Group). May 2001, <http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>
- [9] K. Sycara, M. Paolucci, A. Ankolekar and N. Srinivasan, «*Automated Discovery, Interaction and Composition of Semantic Web services*», Journal of Web Semantics, Volume 1, Issue 1, September 2003, pp. 27-46.
- [10] B. Meyer, «*Applying "Design by Contract"*», in Computer (IEEE), vol. 25, no. 10, October 1992, pages 40-51.
- [11] UDDI: Universal Description, Discovery and Integration, <http://www.uddi.org/specification.html>
- [12] O. K. Zein - *Thèse « Indexation/Découverte et Composition de services distribués »*
Janvier 2005 – ENST Bretagne – département LUSSI
- [13] Kerberos B. Clifford Neuman and Theodore Ts'o. «*Kerberos: An Authentication Service for Computer Networks*», IEEE Communications, 32(9):33-38. September 1994