

## Sujet de recherche

### Influence des contraintes non-fonctionnelles sur le processus décisionnel lors de l'orchestration de web-services

LIP6 – THALES

## Contexte

Le monde de la programmation répartie est en cours de mutation avec les **architectures orientées services** [1]. Ces architectures découplent une application répartie en deux couches. Une couche d'offre de services spécifiés par leurs interfaces et éventuellement des informations de nature sémantique dans un annuaire de services. La seconde couche est la réalisation physique des services sur des machines connectées à Internet et accessibles via des protocoles standards.

**La programmation d'une application est alors vue comme une coordination de haut niveau d'un ensemble de services** dont la composition permet de réaliser le calcul voulu par l'application. Cette coordination, appelée « orchestration », peut se faire au sein d'un **processus** écrit dans une extension du langage BPEL dont l'exécution va d'abord demander de lier les **appels de service abstraits** de l'orchestration à des réalisations de services compatibles fournies par l'annuaire de service.

Il s'agit donc de faire le lien entre une spécification abstraite des fonctionnalités requises par le processus afin de constituer l'application et les fonctionnalités disponibles sur le réseau via leur implémentation concrète sous forme de services web.

## Architectures orientées services

Un système informatique basé sur une architecture orientée services (« Service Oriented Architecture » où SOA) est disponible sous la forme de services réutilisables qu'il est possible de découvrir et composer dynamiquement, avec un couplage lâche. Il se distingue alors des solutions totalement intégrées, de type boîte noire, que sont ERP ou autres progiciels. Cette architecture veut répondre aux besoins de flexibilité et d'adaptabilité rapide et de réutilisabilité mis en avant dans les grands systèmes logiciels actuels et souhaités par les entreprises.

Une des caractéristiques importantes des architectures orientées services tient dans le fait qu'elles reposent sur des protocoles et des représentations de données standards, le plus souvent utilisant des langages en XML.

## Spécification formelle des besoins du processus web

Dans la plupart des processus web le lien avec les services web est fixé de manière statique via l'indication de l'emplacement physique des services sur le réseau. Cela implique la connaissance préalable des services disponibles au moment de l'écriture du processus.

Afin de palier à ce manque de souplesse et s'aligner par rapports aux besoins du domaine d'application, on envisage ici l'utilisation d'une spécification abstraite des besoins on se basant sur les travaux ayant faits leurs preuves dans le monde du Web Sémantique [2] : la représentation des concepts du domaine et de sa sémantique via la définition d'ontologies spécifiques ainsi que l'utilisation de ces concepts pour la définition des besoins du processus web et celle des fonctionnalités offertes par les services web [3,4,5,6].

## Domaine d'application

Les domaines d'application des travaux effectués dans le cadre de cette thèse sont les suivants :

- Systèmes de Systèmes (SoS)
- Applications tactiques militaires (QoS, respect des délais).

- Applications civiles dans les systèmes de gestion de crise.
- « Pervasive Computing ».
- Tout domaine dans lequel on est amené à gérer la coordination dynamique de composants avec une contrainte de qualité et un minimum de garanties.

Dans tous ces domaines, la qualité de service et les délais de réaction prennent une importance cruciale. Les architectures orientées services doivent donc être en mesure de répondre à ces différentes exigences. D'autres caractéristiques sont aussi essentielles dans ces applications :

- La mobilité, qui rend l'apparition et la disparition de services très fréquente ;
- La redondance, qui fait en sorte qu'un même service peut à tout instant être rendu par de nombreuses réalisations physiques mais qui ont toutes une certaine probabilité d'être détruites (cette probabilité varie également fortement selon le type et la phase opérationnelle dans laquelle se retrouve le dispositif, engagé ou non) ;
- L'hétérogénéité, un même service peut être rendu par des dispositifs très différents. Par exemple, une imagerie numérique peut être rendue par la caméra à basse résolution intégrée dans le casque d'un fantassin ou encore la caméra à haute résolution d'un char de reconnaissance.

## **Problématique**

Toutes ces caractéristiques font que l'exécution d'un processus BPEL dans un tel contexte doit être en mesure non seulement de prendre en compte les pannes de services et de réseaux, mais il doit prendre également en compte un ensemble de **choix d'exécution faits dynamiquement entre les différents services disponibles en fonction de leur qualité de service (QoS) instantanée** [7].

Plus généralement, ces valeurs de QoS peuvent inclure toutes les propriétés non-fonctionnelles effectives que les services souhaitent publier. Elles s'étendent des propriétés techniques non-fonctionnelles comme la disponibilité ou la bande passante d'un service, jusqu'aux propriétés non-fonctionnelles de haut niveau liées au domaine de compétence d'un service donné.

## **Objectifs de recherche**

Dans cette thèse, nous proposons de développer un interprète BPEL capable non seulement de faire la liaison dynamique d'un appel abstrait de service à sa réalisation physique (pour obtenir une grande robustesse intrinsèque par rapport aux pannes et aux déconnexions), mais également de gérer dynamiquement la qualité de service instantanée offerte par les différents services. En parallèle, nous proposerons des extensions au langage BPEL qui permettront au programmeur d'exprimer ses politiques de décision dans la liaison et l'utilisation des services physiques en fonction des informations de qualité de service disponible.

Le second défi de cette thèse sera de proposer au programmeur des abstractions de programmation pour le niveau contrôle qui soient faciles d'utilisation et bien adaptées au domaine d'application.

## **Approche proposée**

L'approche proposée pour atteindre le premier de ces objectifs consistera à utiliser une architecture à deux niveaux :

- **Un niveau d'exécution et de composition des services.** La composition est l'étape par laquelle les services disponibles sur le réseau sont sélectionnés et associés au processus web.
- **Et un niveau de monitoring, de décision et de contrôle du niveau d'exécution en fonction des contraintes non fonctionnelles (QoS,...).** C'est ce niveau qui à la charge de prendre les décisions optimales afin de mener à bien le processus web en fonction des QoS requises et de celles fournies par les services sélectionnés.

Nous suivons en cela l'approche des langages réflexifs, qui découpent une telle architecture en un niveau de base (exécution) et un niveau méta de contrôle du niveau de base.

La contribution principale apportée par cette thèse au domaine de la composition consistera à mettre en place une approche en plusieurs étapes capable à différents moments (statique, dynamique) de sélectionner des services web en se basant sur leurs propriétés non-fonctionnelles instantanées et sur les contraintes non-fonctionnelles définies dans les processus web. Dans le cadre de cette thèse, on s'attachera tout particulièrement à la gestion des moments de liaison tardifs et à la résolution des problèmes de décision inhérents.

La réalisation de ces objectifs demande cependant d'adapter cette approche au cas réparti et d'y intégrer des notions de contrôle et de décision qui ont peu été étudiés à ce jour. Les processus de décisions pourront en effet faire appel à des approches de la théorie du contrôle assurant une décision optimale lorsque le problème se modélise comme l'optimisation d'un processus de décision markovien commandé [9] ou encore à des approches venant de l'intelligence artificielle lorsqu'il s'agit de fusionner des données complexes pour ensuite les analyser.

En ce qui concerne le second objectif qui consiste à fournir des solutions faciles d'utilisation et adaptées au domaine, nous proposons la définition d'un langage extensible où différents aspects de décision par rapport à la qualité de service pourront être abordés à l'aide de langages spécifiques aux domaines [8] (« Domain-Specific Languages » ou DSL).

## **Positionnement technologique**

Ce travail de thèse s'inscrit dans la lignée des travaux menés conjointement par Thales, Alcatel CIT Research and Innovation, Bull, Nokia, Schneider Electric et Vodafone dans le cadre du projet ITEA<sup>1</sup> S4ALL (« Services for All ») dont le but est d'explorer les solutions techniques et consolider celles préexistantes permettant la mise en place d'un monde de services orientés utilisateurs simples à créer, partager et utiliser.

Mais il est aussi lié à certaines des problématiques rencontrées au cours du projet CARROLL<sup>2</sup> DYONISOS (Inria-Arles, CEA, Thales) dans le domaine des architectures orientées services et des « systèmes de systèmes » dont le but est de permettre la composition dynamique de services et l'exécution de workflows.

## **Références**

- [1] M. P. Papazoglou and D. Georgakopoulos. Service-oriented computing. *Communications of the ACM*, Volume 46, 2003.
- [2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.
- [3] J. Peer. Bringing together Semantic Web and Web services. In *Proceedings of the First International Semantic Web Conference*, Sardinia, Italy, 2002.
- [4] A. Eberhart. Ad-hoc invocation of semantic Web services. In *IEEE International Conference on Web Services (San Diego, California ICWS 2004)*, pages 116–124, June 2004.
- [5] K. Sivashanmugam, K. Verma, A. P. Sheth, and J. A. Miller. Adding semantics to Web services standards. In *Proceedings of the International Conference on Web Services, ICWS '03*, Las Vegas, Nevada, USA, pages 395–401, June 2003.
- [6] A. Patil, S. Oundhakar, and A. Sheth. Semantic annotation of Web services. Technical report, LSDIS Lab, Department of Computer Science, University of Georgia, March 2003.
- [7] J. Cardoso, A. Sheth, J. Miller, J. Arnold, and K. Kochut. Quality of service for workflows and Web service processes. *Journal of Web Semantic*, 2004.
- [8] Marjan Mernik, Jan Heering, and Anthony M. Sloane. When and how to develop domain-specific languages. *ACM Computing Surveys*, 37(4):316–344, 2005.
- [9] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York, 1994.

---

<sup>1</sup> <http://www.itea-office.org/>

<sup>2</sup> <http://www.carroll-research.org/fr/index.htm>