

Projet ANR-07-JCJC-0155

**Pervasive Service Composition
(PERSO)**

Programme Jeunes-Chercheurs / Jeunes-Chercheuses 2007

A IDENTIFICATION.....	2
B RÉSUMÉ CONSOLIDÉ PUBLIC.....	2
B.1 Résumé consolidé public en français.....	2
B.2 Résumé consolidé public en anglais.....	3
C MÉMOIRE SCIENTIFIQUE.....	5
C.1 Résumé du mémoire.....	5
C.2 Approche scientifique et technique.....	5
C.3 Résultats obtenus.....	6
C.3.1 Niveau comportemental (adaptation, composition et exécution).....	6
C.3.2 Niveau non-fonctionnel (sélection et composition).....	7
C.3.3 Niveau non-fonctionnel (évaluation).....	7
C.3.4 Niveau sémantique (adaptation et composition).....	8
C.3.5 Vers un cadre et une plate-forme d'intégration.....	8
C.3.6 Composition dynamique (détection, recomposition et réparation).....	9
C.4 Exploitation des résultats.....	9
C.5 Discussion et conclusions.....	9
C.6 Références.....	10
D LISTE DES LIVRABLES.....	10
E IMPACT DU PROJET.....	10
E.1 Indicateurs d'impact.....	10
E.2 Liste des publications et communications.....	11
E.3 Liste des éléments de valorisation.....	13
E.4 Bilan et suivi des personnels recrutés en CDD (hors stagiaires)	16

A IDENTIFICATION

Acronyme du projet	PERSO
Titre du projet	Pervasive Service Composition
Coordinateur du projet (société/organisme)	INRIA
Période du projet (date de début – date de fin)	01/11/2007 - 31/10/2010
Site web du projet, le cas échéant	http://anr-perso.ibisc.fr/

Rédacteur de ce rapport	
Civilité, prénom, nom	M. Pascal POIZAT
Téléphone	01 3963 5869
Adresse électronique	pascal.poizat@lri.fr
Date de rédaction	07/12/10

Si différent du rédacteur, indiquer un contact pour le projet	
Civilité, prénom, nom	
Téléphone	
Adresse électronique	

Liste des partenaires présents à la fin du projet (société/organisme et responsable scientifique)	Les projets JCJC non sont pas multi-partenaires (pas de consortium). Le porteur est l'équipe-projet INRIA/ARLES. Cependant, les membres du projet sont de différentes équipes : INRIA/ARLES, Univ. Evry/IBISC, Univ. Paris Dauphine/LAMSADE, Univ. Paris-Est Créteil/LACL*, Univ. Paris-Sud/LRI*, Univ. Pierre et Marie Curie/LIP6* (* - équipes ajoutées en cours de projet en raison de la mobilité ou de la promotion des enseignants-chercheurs du projet).
---	---

B RÉSUMÉ CONSOLIDÉ PUBLIC

B.1 RÉSUMÉ CONSOLIDÉ PUBLIC EN FRANÇAIS

Satisfaire les besoins des utilisateurs par composition automatique de services

Composition de services Web pour et par l'utilisateur final

Le développement d'Internet et de l'informatique diffuse permet d'envisager l'accès à un nombre toujours plus grand de données et d'applications de façon intuitive, agréable et naturelle. L'ingénierie logicielle promeut actuellement la construction d'applications à forte valeur ajoutée à partir de briques logicielles réutilisables, faiblement couplées et distribuées appelées services et pouvant être décrites à quatre niveaux (typage, protocoles d'usage, non-fonctionnel/qualité de service, sémantique) et selon deux points de vue (opérations utilisables par une application ou fonctionnalités compréhensibles par un utilisateur final). Le projet PERSO vise à permettre la composition automatique de services disponibles dans l'environnement des utilisateurs à partir de la description de leurs besoins. Pour cela, il importe d'abord de prendre en compte les différents niveaux de description des services et de développer des mécanismes permettant d'adapter ces derniers en cas d'incompatibilités. Au delà, il est primordial de supporter le caractère distribué des compositions de services et la possibilité de changement à l'exécution des besoins des utilisateurs ou des services disponibles.

Une approche combinant ingénierie à base de modèles et méthodes formelles

Le projet PERSO combine ingénierie dirigée par les modèles et méthodes formelles. L'ingénierie dirigée par les modèles augmente la flexibilité et la généralité des solutions proposées. Les langages normalisés de description de services sont abstraits par des modèles servant d'entrée ou de résultat aux algorithmes développés. Le processus de satisfaction du besoin d'un utilisateur est alors réalisé en trois étapes : l'obtention de modèles à partir de langages permettant la description du besoin et des services disponibles, des algorithmes de composition opérant sur ces modèles et pour finir l'implantation (codage) d'une composition à partir d'un modèle. Les méthodes formelles, utilisées dans la seconde étape, permettent de combiner le caractère rigoureux des algorithmes et la possibilité de les implanter au sein de solutions logicielles. Nous avons plus particulièrement utilisé des méthodes à base de formalismes comportementaux (automates, réseaux de Petri) ou de planification. Une spécificité du projet PERSO est aussi d'intégrer au sein d'une démarche de génie logiciel des techniques plus « exotiques » telles que l'optimisation combinatoire ou l'analyse de performance des réseaux.

Résultats majeurs

De nouvelles solutions ont été développées permettant la sélection, la composition, l'adaptation et l'évaluation de services et de compositions de services. Ces solutions supportent les différents niveaux d'abstraction et de description des services et des besoins des utilisateurs. Une proposition de langage d'exécution de compositions de services, JCWS, plus léger que le standard WS-BPEL a été faite. Le caractère dynamique des services disponibles et des besoins des utilisateurs a aussi été prise en compte au travers de techniques de recomposition et de réparation.

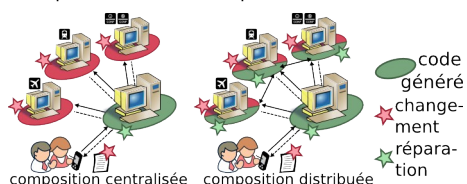
Production scientifique

Cinq articles ont été publiés dans des revues internationales et une quinzaine dans des conférences internationales. Un chapitre sur la composition de services à base d'informations transactionnelles a été publié dans un livre sur les propriétés non-fonctionnelles des systèmes orientés-services. Ces documents couvrent l'ensemble de nos solutions en termes de sélection, composition, adaptation et évaluation de services. Nos travaux ont aussi fait l'objet de plusieurs séminaires invités.

composition à base de modèles



composition & re-composition à l'exécution



Le projet PERSO est un projet de recherche fondamentale coordonné par l'équipe-projet ARLES de l'INRIA. Il associe aussi le laboratoire IBISC (Université d'Évry Val d'Essonne) et le laboratoire LAMSADE (Université Paris Dauphine). Le projet a débuté en novembre 2007 pour une durée de 36 mois. Il a bénéficié d'une aide de l'ANR de 133 464 € pour un coût global de 669 644 €.

B.2 RÉSUMÉ CONSOLIDÉ PUBLIC EN ANGLAIS

Satisfy User Needs through Automatic Service Composition

Web Service Composition for the End-User

The development of the Internet and pervasive computing gives access to an increasing number of data and applications in an intuitive, pleasant, and natural way. In order to build value-added applications, software engineering currently promotes the composition of services, reusable, loosely coupled, and distributed software bricks. Such services can be described at four different, but complementary, description levels (typing, protocols, non-functional/quality of service, semantics) and with two different viewpoints (operations, adequate for being called from an application, and capability, understandable by an end-user). The PERSO project aims at supporting the automatic composition of services available in the end-user's environment in order to fulfill his/her needs. Towards this objective, it is important to take into account the different service description levels and to develop techniques to adapt these services in case of mismatch. Further, it is essential to support the distributed nature of service compositions and possible change in service availability and end-user needs at run-time.

An Approach Combining Model-Driven Engineering and Formal Methods

The PERSO project combines model-driven engineering and formal methods. Model-driven engineering increases the flexibility and genericity of our solutions. Standard or extended service description languages are transformed from/to models which act as inputs or outputs of the developed algorithms. The satisfaction of the end-user needs is then achieved in three steps. Models are first retrieved from languages that enable the description of user needs and available services. Second, composition algorithms operate on these models and yield a composition model. Eventually, model transformation is used to implement (encode) the composition model in some service composition execution language. Formal methods, used in the second step, enable one to combine the rigor of algorithms and the possibility to implement them. More particularly, we have relied on behavioral formal models (automata, Petri nets) and AI planning. A specificity of the PERSO project is also to integrate, within a usual software engineering process, more « exotic » techniques such as combinatorial optimizing or network performance analysis.

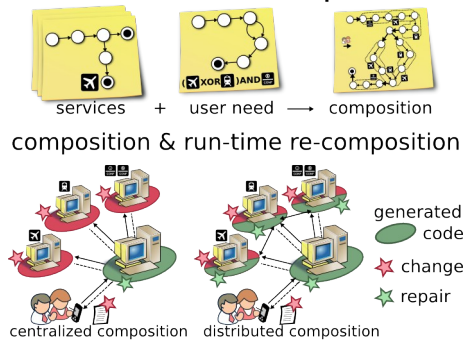
Main Results

New solutions have been developed for the selection, composition, adaptation and evaluation of services and service compositions. This includes taking into account a dynamic context where available services and user needs may change over time. This is particularly supported with recomposition and repair techniques. Different abstraction and description levels are supported for both services and the specification of the end-user needs. A lightweight service composition execution language, JCWS, has been proposed as an alternative to the WS-BPEL standard.

Scientific Outcomes

Our solutions for service selection, composition, adaptation, and evaluation have been presented in five articles in international journals and around fifteen papers in international conferences. The outcomes of the project include a survey chapter on service composition based on transactional information in a book on the non-functional properties of service-oriented systems. Finally, our work has also been presented in several invited talks or seminars.

model-based composition



The PERSO project is a fundamental research project coordinated by INRIA project-team ARLES. The project associates the IBISC (University of Evry Val d'Essonne) and LAMSADE (University Paris Dauphine) labs. It started on November, 2010, for a duration of 36 months and has been supported by a € 133,464 grant from ANR (total cost: € 669,644).

C MÉMOIRE SCIENTIFIQUE

Mémoire scientifique confidentiel : non

Nous souhaitons cependant que le mémoire ne soit pas transféré de HAL vers Arxiv/CoRR de façon à ne pas être indexé dans DBLP.

C.1 RÉSUMÉ DU MÉMOIRE

Le développement d'Internet et de l'informatique diffuse permet d'envisager l'accès à un nombre toujours plus grand de données et d'applications de façon intuitive, agréable et naturelle. L'ingénierie logicielle promeut actuellement la construction d'applications à forte valeur ajoutée à partir de briques logicielles réutilisables, faiblement couplées et distribuées appelées services et pouvant être décrites à quatre niveaux (typage, protocoles d'usage, non-fonctionnel/qualité de service, sémantique) et selon deux points de vue (opérations utilisables par une application ou fonctionnalités compréhensibles par un utilisateur final). Le projet PERSO vise à permettre la composition automatique de services disponibles dans l'environnement des utilisateurs à partir de la description de leurs besoins. Pour cela, il importe d'abord de prendre en compte les différents niveaux de description des services et de développer des mécanismes permettant d'adapter ces derniers en cas d'incompatibilités. Au delà, il est primordial de supporter le caractère distribué des compositions de services et la possibilité de changement à l'exécution des besoins des utilisateurs ou des services disponibles.

Le projet PERSO combine ingénierie dirigée par les modèles et méthodes formelles. Une spécificité du projet PERSO est aussi d'intégrer des techniques plus « exotiques » telles que l'optimisation combinatoire ou l'analyse de performance des réseaux. De nouvelles solutions ont été développées pour la sélection, la composition, l'adaptation et l'évaluation de services et de compositions de services. Elles supportent les différents niveaux d'abstraction et de description des services et des besoins des utilisateurs. Le caractère dynamique des services disponibles et des besoins des utilisateurs a été pris en compte au travers de techniques de recomposition et de réparation.

C.2 APPROCHE SCIENTIFIQUE ET TECHNIQUE

Le projet PERSO se base sur une approche à base d'ingénierie dirigée par les modèles et de modèles formels. L'ingénierie dirigée par les modèles augmente la flexibilité et la généralité des solutions proposées. Les langages normalisés de description de services

sont abstraits par des modèles servant d'entrée ou de résultat aux algorithmes développés. Le processus de satisfaction du besoin d'un utilisateur est alors réalisé en trois étapes : l'obtention de modèles à partir de langages permettant la description du besoin et des services disponibles, des algorithmes de composition opérant sur ces modèles et pour finir l'implantation d'une composition à partir d'un modèle. Les modèles formels permettent de combiner le caractère rigoureux des algorithmes et la possibilité de les implanter. Un second élément de l'approche utilisée dans le projet est la validation des concepts et des techniques proposés par l'utilisation d'études de cas, la simulation et le développement de prototypes.

Les modèles formels utilisés sont de différentes natures, adaptés aux spécificités des niveaux de description des services. On trouvera ainsi des automates (LTS, STS) et des workflows (ou réseaux de Petri) pour la modélisation des aspects comportementaux et conversationnels. Les automates sont privilégiés dans le cas de descriptions au niveau des opérations fournies/requises, tandis que les workflows sont plutôt utilisés en cas de descriptions en termes d'activités ou de fonctionnalités. On trouvera aussi des modèles spécifiques dans le cas des aspects non-fonctionnels, comme les automates stochastiques. Enfin, les aspects sémantiques utilisent intensivement des descriptions à base d'ontologies. Côté techniques, la discrimination se fait plutôt en fonction de l'objectif visé (sélection, adaptation/composition, évaluation) que du niveau d'interface. Ainsi, on trouvera l'utilisation de techniques d'optimisation combinatoire pour la sélection, de model-checking et de planification pour l'adaptation et la composition, et de techniques d'analyse de performance pour l'évaluation des compositions.

C.3 RÉSULTATS OBTENUS

C.3.1 NIVEAU COMPORTEMENTAL (ADAPTATION, COMPOSITION ET EXÉCUTION)

La prise en compte des protocoles des services lors de leur composition met en exergue de possibles incompatibilités, les services étant développés par différents tiers. Le principal verrou abordé au niveau comportemental est la construction de compositions présentant un caractère adaptatif qui permet la réconciliation de protocoles partiellement incompatibles. Nous avons défini un mécanisme d'adaptation logicielle assurant la compatibilité de protocoles par construction d'un adaptateur centralisé [1]. Formellement, étant donné n protocoles L_i (systèmes de transitions étiquetées, LTS), il s'agit d'engendrer un LTS A , tel que la mise en parallèle de A avec les L_i soit exempte de blocages. Pour ce faire, le processus d'adaptation utilise une spécification abstraite des correspondances entre messages échangés (contrat d'adaptation). Ce processus est à la fois restrictif (il supprime les messages conduisant à des blocages) et génératif (il peut ré-ordonner les messages). Par ailleurs, il peut résoudre les incompatibilités entre n protocoles et n'est pas limité à la compatibilité 2-à-2 comme les travaux existants d'adaptation de services. Nous avons ensuite amélioré notre approche par des techniques de construction d'adaptateurs à-la-volée qui évitent le calcul coûteux de produits complets de LTS [11]. Nous avons aussi appliqué notre approche à la construction d'orchestrateurs-adaptateurs en faisant le lien entre approche basée modèle et langages de services en prenant en compte les phases aval (traduction de WS-BPEL en LTS) et amont (traduction de LTS en WS-BPEL) [11]. Ces différents résultats prennent place lors de la phase de composition, après la découverte des services. Ils adoptent un point de vue opérationnel sur les services et le besoin utilisateur y est pris en compte de façon banalisée, en tant que l'un des protocoles à composer/adapter.

Le langage standard actuel de composition centralisée (orchestration) de services Web est WS-BPEL. Un problème de WS-BPEL est sa relative lourdeur. Nous avons proposé une alternative à l'orchestration en WS-BPEL sous la forme d'une extension du langage Java avec des primitives inspirées de WS-BPEL, JCWS [2]. JCWS permet de considérer les sous-services utilisés dans une composition comme des ressources Java (classes offrant des opérations), ils peuvent donc être inclus dans une composition de façon similaire aux paquetages (sous la forme de l'URL de leur interface WSDL). Un avantage de JCWS est

son efficacité par compilation des services en code Java pur déployé par utilisation de la plate-forme AXIS.

C.3.2 NIVEAU NON-FONCTIONNEL (SÉLECTION ET COMPOSITION)

Nous avons proposé un algorithme de sélection de service (permettant la réalisation d'une composition définie en termes de workflow) qui satisfait les préférences utilisateur exprimées sous la forme de critères de QoS pondérés ainsi que sous la forme de niveaux de risque transactionnels [10,3]. En raison du domaine applicatif du projet (informatique diffuse, sans ou avec peu d'infrastructure, et équipements pauvres en ressources), nous avons ciblé un ensemble donné d'attributs de QoS : disponibilité, temps de réponse (moyen, pire) et abstraction de ressources. Nous avons étendu le formalisme des réseaux de Petri colorés (CPN) pour incorporer des propriétés non-fonctionnelles dans la composition de services. Nous avons présenté un algorithme de sélection basé CPN qui satisfait à la fois les besoins fonctionnels (entrées/sorties) et non-fonctionnels (critères de QoS pondérés et niveau de risque transactionnel) des utilisateurs. Un premier résultat est un CPN qui représente une composition de service transactionnelle dont les sous-services optimisent (chacun) localement la QoS [16,4]. Ces propositions optimisent la QoS localement. Dans une première tentative de proposer un algorithme satisfaisant les propriétés non-fonctionnelles globalement, nous avons restreint les attributs de QoS à la durée et au prix d'exécution. Nous avons alors défini un algorithme de composition au niveau QoS qui maximise la satisfaction d'un utilisateur (en temps et en prix) tout en minimisant le regret associé à l'iniquité entre utilisateurs [25]. Nous avons enfin réalisé une étude sur l'application des techniques d'optimisation à la sélection basée QoS de services [30].

C.3.3 NIVEAU NON-FONCTIONNEL (ÉVALUATION)

Nos premiers résultats concernent l'évaluation de performance des constructeurs du langage standard d'orchestration de services, WS-BPEL. Nous avons proposé des formules analytiques pour le calcul du temps de réponse d'un service Web composite. Plus précisément, nous avons calculé le temps de réponse des différents constructeurs directement supportés par WS-BPEL. La démarche proposée permet le calcul exact du temps moyen de réponse. Contrairement aux précédentes études, nous avons considéré le fait que l'un des objectifs des architectures orientées services est de permettre l'interopérabilité entre systèmes de plus en plus hétérogènes. En effet, nous avons considéré le cas où les temps de réponse des différents services élémentaires participant à une composition donnée sont de loi exponentielle [9]. Nous avons aussi pris en compte dans [18] les résultats de nombreuses études expérimentales montrant que la loi « heavy-tailed » (Pareto, par exemple) est celle qui ajuste au mieux les temps de réponse des services Web élémentaires.

Nous nous sommes aussi intéressés au calcul des bornes du temps de réponse des services Web composites en proposant des modèles permettant d'améliorer la complexité de calcul de $O(n^2)$ à $O(n\sqrt{n})$. Nous avons étudié une famille de patrons de composition ne pouvant pas s'exprimer en utilisant les constructeurs de WS-BPEL. L'intérêt de notre approche provient du fait que l'on peut trouver un compromis entre la qualité de la borne et le temps de son calcul numérique. Dans le cas de services élémentaires hétérogènes, nous avons proposé des modèles bornants qui fournissent des bornes supérieures et inférieures sur les temps de réponse d'un modèle de service Web composite dans lequel le problème de l'explosion combinatoire du nombre d'états est posé [19].

Nous avons proposé une extension de l'architecture conventionnelle des services Web, en prenant en compte différents critères de qualité de service dans la découverte et la sélection des meilleurs services [20]. Plus précisément, nous avons proposé un cadre multicritère pour la sélection de la meilleure composition d'un service Web. Ce cadre étend l'architecture conventionnelle des services Web par l'ajout dans le registre UDDI d'un nouveau composant, MEC (Multicriteria Evaluation Component).

Enfin, nous avons proposé de nouvelles techniques de simplification pour le calcul du temps d'exécution de bout en bout pour un modèle de service Web. Le but est de vérifier si le temps d'exécution ne dépasse pas un seuil (time-out) [5].

C.3.4 NIVEAU SÉMANTIQUE (ADAPTATION ET COMPOSITION)

La composition adaptative comportementale requiert la définition de contrats d'adaptation. Pour lever cette contrainte, nous avons proposé une approche entièrement automatique qui substitue à ces contrats des annotations sémantiques associées aux données échangées entre services [8]. Un autre point clé de cette approche est la construction d'orchestrateurs-adaptateurs locaux, associés aux services et non-plus centralisés. Cette approche s'intéresse au niveau d'abstraction opérationnel.

A contrario, nous avons proposé dans [12] une technique de composition automatique de service sémantiques à base de planification (graphplan) au niveau activités. Elle permet la prise en compte non seulement d'incompatibilités entre données produites/nécessaires au chainage des services utilisés, mais aussi la différence de niveau d'abstraction entre le besoin utilisateur (supposé plus abstrait) et les capacités sémantiques des services disponibles. Dans le but de permettre l'intégration des mécanismes de composition (à différents niveaux et avec la dichotomie dans les descriptions à base d'opérations ou d'activités/capacités) nous avons étendu ce travail par la prise en compte de modèles plus riches (opérations et capacités pour les services, comportement et sémantique pour les services et pour les besoins) [21]. Nous avons aussi étudié l'utilisation de mécanismes de raisonnement et de structures d'événements pour améliorer les processus de composition de services [34].

C.3.5 VERS UN CADRE ET UNE PLATE-FORME D'INTÉGRATION

Le projet PERSO a conduit à la définition d'algorithmes de sélection et de composition à différents niveaux de description pour les interfaces des services et pour les besoins des utilisateurs finaux. La plupart de ces solutions ont été validées par la réalisation de prototypes (voir plus bas). De plus, des modèles similaires (à quelques caractéristiques liées au niveau de description près) sont utilisés en entrée et sortie de ces algorithmes et prototypes: fichiers au format WSDL pour les interfaces de base, STS ou workflows pour les interfaces comportementales, etc. Nous avons donc réalisé une version préliminaire de cadre permettant l'intégration de nos différents résultats. Ainsi nous avons défini un format textuel standard pour les STS et pour les workflows (ainsi que les parsers adéquats). L'idée est alors de déployer les différents algorithmes de sélection et de composition eux-mêmes en tant que services Web. Ceci a pour l'instant été expérimenté avec un algorithme-service de composition comportemental+sémantique (pycompose [21]) et deux algorithmes-services de génération de code à partir d'un modèle de composition (STS2BPEL et WF2BPEL). Nous avons défini un format permettant de spécifier un processus de composition (sorte de « makefile » enchaînant les algorithmes définis dans le projet). Nous avons ensuite défini un service Web composite supportant l'exécution de n'importe quel processus de composition. Ce service reçoit la description du processus ainsi que les entrées nécessaires à son exécution (par exemple la description des besoins et les URL des interfaces de services à utiliser). Il se charge alors d'invoquer en séquence (seul opérateur supporté pour l'instant dans les processus de composition) différents algorithmes-services et termine en renvoyant le résultat du dernier. Nous avons expérimenté l'appel à ce service composite depuis un petit équipement personnel (iPad) sur lequel l'utilisateur spécifie son besoin et le processus puis se voit retourner le code WS-BPEL de la composition. Ce travail demanderait à être poursuivi et finalisé, entre autres autour de la question de « bonnes » interfaces pour permettre la saisie de ses besoins à l'utilisateur (et une transparence aux algorithmes utilisés). A terme, ce travail représenterait une instanciation du principe « software as a service » du « cloud computing ».

C.3.6 COMPOSITION DYNAMIQUE (DÉTECTION, RECOMPOSITION ET RÉPARATION)

D'un point de vue non-fonctionnel, nous avons étudié le problème de la (re)sélection/ (re)composition de service dans un cadre contextuel, c'est-à-dire dans lequel les contextes en termes d'utilisateur(s) et de services disponibles évoluent en raison par exemple de la mobilité des utilisateurs ou de services occupés ou indisponibles. Nous avons proposé une méthode de sélection de service de remplacement qui optimise l'interaction entre propriétés non-fonctionnelles locales et globales (QoS et propriétés transactionnelles) [33].

Sur la base de notre modèle de composition au niveau sémantique par planification [12] nous avons proposé un algorithme de réparation de plans incomplets permettant une alternative à la re-composition complète en cas de disparition de service utilisé ou d'apparition de nouvel objectif utilisateur [17]. L'algorithme permet l'obtention de compositions mises à jour de meilleure qualité et de façon plus rapide que la re-composition. Cet algorithme a été modifié dans [22,23] de façon à augmenter le taux de réussite en cas de disparition de service sans remplaçant(s).

De façon complémentaire à ces travaux permettant la correction des problèmes, il est indispensable de définir des techniques permettant de les détecter et de les isoler (diagnostic et diagnosticabilité). Nous avons montré dans [24] que les définitions et techniques de vérification habituelles de la diagnosticabilité ne pouvaient s'appliquer aux compositions de services Web. Nous avons proposé une généralisation de ces définitions à base d'une structure de jeux et une généralisation des techniques de vérification de diagnosticabilité par l'utilisation d'un fragment de la logique temporelle de temps alterné [24]. Sur ces bases, nous avons introduit les notions de diagnosticabilité intrinsèque (« context-free diagnosability ») et de diagnosticabilité en fonction d'un contexte de composition (« context-aware diagnosability ») [24]. Cette dernière notion permet de répondre à question de savoir si un service Web est diagnosticable dans le contexte d'une composition (centralisée ou non).

C.4 EXPLOITATION DES RÉSULTATS

Nos résultats ont donné lieu à des publications (disponibles sur <http://anr-perso.ibisc.fr/>) mais aussi à la réalisation de prototypes permettant d'évaluer les solutions proposées, qu'il s'agisse de sélection (TQoSWS), d'adaptation et de composition (compositor+scrutator, pycompose) ou de réparation dynamique de compositions (PGA). Un point fort du projet PERSO est l'utilisation d'une approche à base de modèles. Ainsi, plusieurs outils ont été développés pour faire le lien entre langages de description de services et modèles (BPEL2STS) ou entre modèles de composition et langages d'implantation et d'exécution (STS2BPEL, WF2STS). Ces derniers sont utilisables hors du contexte particulier (la composition) du projet PERSO. Les compétences développées dans le projet PERSO seront exploitées dans le cadre du projet PIMI (Personal Information Management through Internet), sélectionné à l'appel ANR VERSO 2010.

C.5 DISCUSSION ET CONCLUSIONS

Le projet PERSO était relativement ambitieux pour un projet mono-partenaire (jeunes-chercheurs), en visant à la fois le développement de modèles, d'algorithmes, et de prototypes pour la sélection, la composition, l'adaptation et l'analyse de services et de compositions de services; mais aussi l'application de ces algorithmes et prototypes au domaine de l'informatique diffuse. Cet aspect a été principalement abordé sous l'angle de contraintes du domaine: distribution (orchestration vs chorégraphie), dynamisme (réparation de compositions) et efficacité (techniques d'adaptation et de composition efficaces, techniques d'évaluation de la qualité des compositions, langage d'orchestration alternatif à WS-BPEL, réflexion sur le requêtage et l'utilisation de compositions via des équipements mobiles personnels). Une campagne d'expérimentation dans un contexte d'informatique diffuse reste à faire.

Nos résultats en termes d'approche à base de modèles et de composition soit par un concepteur/architecte logiciel soit par l'utilisateur final a de possibles applications dans le cadre industriel: par l'apport de techniques de construction sûres et automatiques de compositions de services (SSII spécialistes SOA) et pour la construction de « clouds » à forte valeur ajoutée (permettant la génération de compositions à la demande, « software as a service »). L'impact sociétal est encore limité. Une vraie réflexion sur les « bonnes » interfaces et langages permettant aux utilisateurs de spécifier leurs besoins doit être menée. Nos modèles et langages restent d'un accès assez limité (automates et workflows étant cependant usuels en industrie). Le projet PIMI qui commence et qui comprend des partenaires industriels et une activité sur les interfaces et l'évaluation par les utilisateurs, pourrait permettre d'évaluer l'impact industriel et sociétal de nos résultats.

C.6 RÉFÉRENCES

En raison de la taille limitée (5 pages), les références sont celles de la section E.2 du compte-rendu de fin de projet disponible ici : <http://anr-perso.ibisc.fr/ANRdocuments>

D LISTE DES LIVRABLES

Aucun livrable. Cependant, nous avons rédigé deux documents présentant les différents langages existants ainsi que les modèles formels et les principaux travaux portant sur la composition à ces niveaux d'interface: <http://anr-perso.ibisc.fr/PublicDocuments>

E IMPACT DU PROJET

E.1 INDICATEURS D'IMPACT

Nombre de publications et de communications (à détailler en E.2)

Les projets Jeunes-chercheurs/Jeunes-chercheuses sont mono-partenaires.

		Publications multipartenaires	Publications monoparttenaires
International	Revue à comité de lecture	n/a	5
	Ouvrages ou chapitres d'ouvrage	n/a	1
	Communications (conférence)	n/a	18
France	Revue à comité de lecture	n/a	0
	Ouvrages ou chapitres d'ouvrage	n/a	0
	Communications (conférence)	n/a	1
Actions de diffusion	Articles vulgarisation	n/a	0
	Conférences vulgarisation	n/a	5
	Autres	n/a	4

Autres valorisations scientifiques (à détailler en E.3)

	Nombre, années et commentaires (valorisations avérées ou probables)
Brevets internationaux obtenus	0

Brevet internationaux en cours d'obtention	0
Brevets nationaux obtenus	0
Brevet nationaux en cours d'obtention	0
Licences d'exploitation (obtention / cession)	0
Créations d'entreprises ou essaimage	0
Nouveaux projets collaboratifs	1 projet ANR appel VERSO 2010
Colloques scientifiques	0
Autres (préciser)	9 prototypes 2 nouveaux partenariats

E.2 LISTE DES PUBLICATIONS ET COMMUNICATIONS

International / Revues à comité de lecture

- [1] Carlos Canal, Pascal Poizat and Gwen Salaün. *Model-based Adaptation of Behavioural Mismatching Components*. in IEEE Transactions on Software Engineering, 34(4):546-563, 2008
- [2] Céline Boutrous-Saab, Demba Coulibaly, Serge Haddad, Tarek Melliti, Patrice Moreaux, and Sylvain Rampacek. *An Integrated Framework for Web Services Orchestration*. in International Journal of Web Services Research, 6(4), 1-29, 2009.
- [3] Joyce El Haddad, Maude Manouvrier, and Marta Rukoz. *TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition*. In IEEE Transactions on Services Computing, 3(1):73-85. 2010
- [4] Yudith Cardinale, Joyce El Haddad, Maude Manouvrier, and Marta Rukoz. *CPN-TWS: A Colored Petri-Net Approach for Transactional-QoS driven Web Service Composition*. In International Journal of Web and Grid Services, to appear, 2011. (acceptation letter available)
- [5] Jean-Michel Fourneau, Lynda Mokdad, and Nihal Pekergin. *Stochastic bounds for performance evaluation of Web services*. In Journal of Concurrency and Computation : Practice and Experience, 22(10):1286-1307, 2010.

International / Ouvrages ou chapitres d'ouvrages

- [6] Yudith Cardinale, Joyce El Haddad, Maude Manouvrier, and Marta Rukoz. *Transactional-aware Web Service Composition: A Survey*. in Handbook of Research on Non-Functional Properties for Service-Oriented Systems: Future Directions, to appear, 2011. (acceptation letter available)

International / Conférences et ateliers (workshops)

- [7] Manel Fredj, Nikolaos Georgantas, Valerie Issarny, and Apostolos Zarras. *Dynamic Service Substitution in Service-Oriented Architectures*. in SERVICES'08 - IEEE Congress on Services, pp 101-104, IEEE, 2008
- [8] Tarek Melliti, Pascal Poizat and Sonia Ben Mokhtar. *Distributed Behavioural Adaptation for the Automatic Composition of Semantic Services*. in FASE'2008 - Fundamental Approaches to Software Engineering. Lecture Notes in Computer Science (LNCS) 4961:146-162, Springer, 2008
- [9] Serge Haddad, Lynda Mokdad, and Samir Youcef. *Response Time Analysis for Composite Web Services*. in CSNDSP'2008 - IEEE Symposium on Communication Systems, Networks and Digital Signal Processing. pp 1-5, 2008

- [10] Joyce El Haddad, Maude Manouvrier, Guillermo Ramirez, and Marta Rukoz. *QoS-driven Selection of Web Services for Transactional Composition*. in ICWS'08 - IEEE International Conference on Web Services, pp 653-660. 2008
- [11] Radu Mateescu, Pascal Poizat, and Gwen Salaün. *Adaptation of Service Protocols using Process Algebra and On-the-Fly Reduction Techniques*. in ICSOC'08 - International Conference on Service Oriented Computing. Lecture Notes in Computer Science (LNCS) 5364:84-99, Springer, 2008
- [12] Sandrine Beauche, and Pascal Poizat. *Automated Service Composition with Adaptive Planning*. in ICSOC'08 - International Conference on Service Oriented Computing (short papers). Lecture Notes in Computer Science (LNCS) 5364:530-537, Springer, 2008
- [13] Lina Bentakouk, Pascal Poizat, and Fatiha Zaïdi. *A Formal Framework for Service Orchestration Testing based on Symbolic Transition Systems*. in TESTCOM'2009 – 21st IFIP WG 6.1 International Conference on Testing of Communicating Systems. Lecture Notes in Computer Science (LNCS) 5826:16-32, Springer, 2009
- [14] Youssou Kasse, Lynda Mokdad, and Mbaye Sene. *Performance analysis of Web services architecture*. in IEEE workshop on Performance evaluation of communications in distributed systems and Web based service architectures, at ISCC'2009, pp 94 – 98. 2009
- [15] Jalel Ben Othman, Lynda Mokdad, Mohamed Ould Cheikh, and Mbaye Sene. *Performance analysis of composite web services using Stochastic Automata Networks over IP network*. in ISCC'2009 – IEEE Symposium on Computers and Communications, pp 92 – 97. 2009
- [16] Yudith Cardinale, Joyce El Haddad, Maude Manouvrier, and Marta Rukoz. *Web service selection for transactional composition*, in ICCS'2010 – 10th International Conference on Computational Science, Procedia Computer Science, 1(1):2683-2692. 2010
- [17] Yuhong Yan, Pascal Poizat, and Ludeng Zhao. *Repairing Service Compositions in a Changing World*. in Studies in Computational Intelligence, 296:17-36, Springer, 2010. Selected papers from SERA'2010 – 8th ACIS conference on Software Engineering Research, Management & Applications.
- [18] Serge Haddad, Lynda Mokdad, and Samir Youcef. *Response time of BPEL4WS constructors*. In ISCC'10 - IEEE Symposium on Computers and Communications, pp 695-700, 2010.
- [19] Youssou Kasse, Lynda Mokdad, and Samir Youcef. *Stochastic bounds for end-to-end delay in Web service model*. In ISCC'10 - IEEE Symposium on Computers and Communications, pp 1121-1126, 2010
- [20] Serge Haddad, Lynda Mokdad, and Samir Youcef. *Selection of the best composite Web service based on quality of service*. in BPSC'2010 - 3rd International Conference on Business Process and Services Computing. Lectures Notes in Informatics (LNI), 177:255-266, Springer, 2010.
- [21] Pascal Poizat and Yuhong Yan. *Adaptive Composition of Conversational Services through Graph Planning Encoding*. in ISoLA'2010 - International Symposium On Leveraging Applications of Formal Methods, Verification and Validation. Lecture Notes in Computer Science (LNCS), 6416:35-50, Springer, 2010.
- [22] Yuhong Yan, Pascal Poizat, and Ludeng Zhao. *Self-Adaptive Service Composition through Graphplan Repair*. In ICWS'10 - International Conference on Web Services (work in progress papers), pp 624-627, 2010.
- [23] Yuhong Yan, Pascal Poizat, and Ludeng Zhao. *Repairs vs. Recomposition for Broken Service Compositions*. in ICSOC'2010 - International Conference on Service Oriented Computing. Lecture Notes in Computer Science (LNCS), 6470:152-166, Springer, 2010.
- [24] Tarek Melliti and Philippe Dague. *Generalizing diagnosability definition and checking for open systems: a Game structure approach*. In DX'10 - 21th International Workshop on Principles of Diagnosis, 2010.

France / Conférences

- [25] Joyce El Haddad, and Olivier Spanjaard. *Composition de services web et équité vis-à-vis des utilisateurs finaux*. in ROADEF'09 – 10ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Decision, 2009

Actes de diffusion / Conférences vulgarisation (« invited talks », « tutorials », etc.)

- [26] Joyce El Haddad. *QoS-Aware Workflow Realization in a Web Service Context*. Séminaire invité, Université de Málaga, Espagne, mai 2009
- [27] Pascal Poizat. *Testing Web Service Orchestrations*. Séminaire invité, Université de Málaga, Espagne, mai 2009
- [28] Pascal Poizat. *Model-Based Software Adaptation. Introduction and Contributions*. Séminaire lors des journées de rencontre NICTA – LRI/INRIA Saclay – IRISA/INRIA Rennes – LAAS/CNRS, Saclay, France, juillet 2009
- [29] Pascal Poizat. *Model-Based Adaptation. Introduction and Contributions*. Séminaire invité, équipe-projet INRIA ADAM, Lille, janvier 2010.
- [30] Joyce El Haddad. *Optimization Techniques for QoS-Aware Workflow Realization in Web Services Context*. Invited Talk at the Third International Workshop on Resource Discovery (RED2010) joint to the 12th International Conference on Information Integration and Web-based Applications & Services (iiWAS2010), 2010.

Actes de diffusion / Autres

- [31] Iyad Alshabani, Joyce El Haddad, Nikolaos Georgantas, Tarek Melliti, Lynda Mokdad, and Pascal Poizat. *Projet ANR PERSO: Pervasive Service Composition (poster)*. Grand Colloque STIC, 5 – 7 janvier 2010, Paris.
- [32] Iyad Alshabani, Joyce El Haddad, Nikolaos Georgantas, Tarek Melliti, Lynda Mokdad, and Pascal Poizat. *Projet ANR PERSO: Pervasive Service Composition (poster)*. Journées nationales du GDR Génie de la Programmation et du Logiciel (GPL), 10 – 12 mars 2010, Pau.
- [33] Joyce El Haddad, Maude Manouvrier, Stephan Reiff-Marganeic, and Marta Rukoz. *Context-based transactional service selection approach for service composition*. Notes de Recherche n°49, LAMSADE, 2010.
- [34] Paraskevi Zerva. *Automatic Service Composition for Sensor Data*. Master Thesis Report, Université d'Evry Val d'Essonne (ERASMUS with. University of Athens). 107 pages. 2010.

E.3 LISTE DES ÉLÉMENTS DE VALORISATION

Logiciels / Prototypes

Les différentes solutions développées dans le projet ont pour la plupart été validées à l'aide de prototypes de démonstration. Il ne s'agit pas à proprement parler de logiciels vu qu'il ne sont pas packagés et mis en ligne. Ils sont disponibles sur demande et nous savons que des collègues en utilisent ou envisagent d'en utiliser certains (compositor+scrutator développé en collaboration avec l'Université de Malaga est utilisé dans leur plate-forme ITACA présentée à la conférence ICSE'09 ; BPEL2STS va éventuellement être réutilisé dans le cadre du projet ANR USS-SimGrid). Compositor, scrutator, BPEL2STS et STS2BPEL sont présentés dans un article de journal soumis.

BPEL2STS. Traduit une description de conversation de service dans le langage standard d'orchestration, WS-BPEL, en un système de transition étiqueté symbolique (STS). Cette transformation de modèle prend en compte une sous-partie importante du langage WS-BPEL et permet le développement d'algorithmes basés-modèles sur les STS (ex:

compositor). BPEL2STS résulte des travaux présentés dans [11,13]. Contact: pascal.poizat@lri.fr

STS2BPEL. Une fois qu'un modèle de composition (STS) au niveau opérations est construit, il faut déployer et exécuter cette composition. STS2BPEL est une transformation de modèle qui cible le langage standard d'exécution de composition de services centralisée, WS-BPEL. Le code généré comprend le code WS-BPEL mais aussi les descriptions d'interfaces (WSDL) permettant de l'utiliser. Le déploiement est pour l'instant fait manuellement avec un IDE. STS2BPEL est un prototype qui démontre l'intérêt de l'approche à base de modèles (IDM) pour la composition de services. Ainsi, il une perspective est de réutiliser une grande partie du code de STS2BPEL pour générer un code exécutable dans le langage JCWS, substitut à WS-BPEL. STS2BPEL résulte des travaux présentés dans [11]. Contact: pascal.poizat@lri.fr

WF2BPEL. Suit le même principe que STS2BPEL mais permet l'obtention d'un code exécutable en WS-BPEL à partir d'un modèle de composition workflow au lieu de STS. Comme pour STS2BPEL, une perspective est de réutiliser une partie du code de WF2BPEL pour générer du code JCWS. Contact: pascal.poizat@lri.fr

Compositor. Transforme des modèles de composition comportementale de services (LTS pour les services disponibles et pour le besoin utilisateur, contrat d'adaptation entre protocoles) en algèbre de processus (LOTOS). Il est ensuite possible d'utiliser l'atelier CADP pour analyser, simuler, etc. ce problème de composition. Compositor résulte des travaux présentés dans [11]. Contact: pascal.poizat@lri.fr

Scrutator. Permet la génération automatique d'adaptateur / modèles comportementaux de composition de services à partir d'une description en LOTOS d'un problème de composition (générée avec compositor). L'apport de scrutator par rapport à des outils existants et de permettre la suppression à la volée (sans construire totalement d'automate produit global) des suites d'interactions entre services conduisant à des blocages. Cette fonctionnalité peut d'ailleurs être utilisée hors du contexte de la composition de services et a été demandée sur le forum de CADP. Scrutator résulte des travaux présentés dans [11]. Contact: pascal.poizat@lri.fr

TQoSWSS. Permet la sélection de services Web pour réaliser un besoin utilisateur exprimé sous forme de workflow tout en garantissant localement la meilleure qualité de service et globalement une propriété transactionnelle. TQoSWSS résulte des travaux présentés dans [3]. Contact: elhaddad@lamsade.dauphine.fr

Pycompose. Transforme des modèles riches pour la composition de service (supportant à la fois les aspects comportementaux et sémantiques, les capacités et les opérations) en problème de planification, résout ce problème à l'aide d'un planificateur (pycompose est générique et permet l'utilisation de nombreux planificateurs) puis en cas de plan-solution trouvé, transforme ce plan en modèle de composition de service (workflow) qui peut être ensuite implanté dans un langage d'orchestration au choix (par exemple en WS-BPEL à l'aide de WF2BPEL). Pycompose résulte des travaux présentés dans [21]. Contact: pascal.poizat@lri.fr

MEC. Ce composant (Multicriteria Evaluation Component) étend le registre de services jUDDI de façon à proposer à l'utilisateur une recommandation basée sur différents critères non-fonctionnels de qualité de services Web ayant une fonctionnalité donnée. Contact : lynda.mokdad@univ-paris12.fr

PGA. Répare une composition de service en cas de disparition de service ou de modification des objectifs de composition. PGA opère au niveau d'un modèle de composition décrit à l'aide d'une structure de planification (par exemple générée à l'aide

de pycompose) appelée graphplan. PGA résulte des travaux présentés dans [17,22,23].
Contact: pascal.poizat@lri.fr

JCWS. Langage d'orchestration de services, JCWS [2] étend Java avec des primitives de communication transformant un programme Java en un service Web. Le code JCWS est compilé en Java pur puis déployé sous AXIS. La gestion des interactions longues est ainsi reléguée aux souches servantes du service. Contact : tarek.melliti@ibisc.fr

Nouveaux projets

Projet PIMI (Personal Information Management through Internet)

projet sélectionné à l'appel ANR VERSO 2010

consortium : GENIGRAPH (PME, porteur), CTIE (Centre des Technologies de l'Information de l'Etat, gouvernement du Luxembourg), Institut Télécom Sud Paris (ex INT), INRIA, IRIT/IHCS, IRIT/LILac, LRI (**membres ANR PERSO**), Montimage (PME).

Les membres du projet PERSO ont été approchés par le porteur (GENIGRAPH) lors du grand colloque STIC organisé par l'ANR en janvier 2010 suite à la présentation d'un poster sur les objectifs et résultats du projet PERSO. Les compétences que nous avons développées sur la composition de services aux niveaux fonctionnel, comportemental, non-fonctionnel et sémantique seront valorisées dans le projet PIMI.

Nouveaux partenariats

Le départ en congé parental d'un membre du projet nous a conduit à la remplacer par un collègue de l'**Université Pierre et Marie Curie** (laboratoire LIP6). A cette occasion nous avons pu travailler avec lui sur les approches à base de modèles de composition distribués (chorégraphie) à l'exécution. Ces travaux ne sont pas terminés mais nous comptons continuer à collaborer sur ce thème.

Lors du projet, nous avons aussi été approchés par une collègue de l'**Université Concordia** (Canada) qui travaillait comme nous sur la composition de services à base de techniques de planification. Nous avons débuté une collaboration avec elle sur les aspects dynamiques de la composition de service (réparation de compositions) qui a donné lieu à plusieurs résultats [17,22,23] et un prototype (PGA). Nous avons soumis un projet en commun dans le cadre de l'appel 2009 du FQRNT (Fonds Québécois de la Recherche sur la Nature et les Technologies) mais il n'a pas été financé. Nous continuerons nos collaborations dans le cadre du projet ANR PIMI (voir plus haut) dans lequel elle est en tant que membre extérieure rattachée pour le projet au partenaire LRI.

	<h2>Compte-rendu de fin de projet</h2>	

E.4 BILAN ET SUIVI DES PERSONNELS RECRUTÉS EN CDD (HORS STAGIAIRES)

Comme demandé, les stagiaires (2 personnes) et les personnels embauchés pour une durée inférieure à trois mois (2 personnes) ne sont pas indiqués (les informations les concernant sont disponibles sur demande).

Identification				Avant le recrutement sur le projet			Recrutement sur le projet				Après le projet				
Nom et prénom	Sexe H/F	Adresse email (1)	Date des dernières nouvelles	Dernier diplôme obtenu au moment du recrutement	Lieu d'études (France, UE, hors UE)	Expérience prof. Antérieure, y compris post-docs (ans)	Partenaire ayant embauché la personne	Poste dans le projet (2)	Durée missions (mois) (3)	Date de fin de mission sur le projet	Devenir professionnel (4)	Type d'employeur (5)	Type d'emploi (6)	Lien au projet ANR (7)	Valorisation expérience (8)
Iyad ALSHABANI	H	iyad.alshabani@gmail.com	07/12/10	Doctorat Informatique	France	7 (inclus post-doc, ATER, ingénieur, vacataire)	INRIA	Post-doc	13	31/01/10	Post-doc France	Enseignement et recherche publique	Chercheur	Non	Oui
Antoine DAUTRICHE	H	antoine.dautriche@gmail.com	26/11/10	Master 1 Informatique	France	0	INRIA	Niveau ingénieur	3	30/09/10	Étudiant en M2 recherche à l'Université d'Evry	n/a	n/a	Oui	Oui