



HAL
open science

Une approche basée SOA pour l'interconnexion de workflows : application au "transfert de cas "

Saida Boukhedouma, Zaia Alimazighi, Mourad Chabane Oussalah, Dalila Tamzalit

► To cite this version:

Saida Boukhedouma, Zaia Alimazighi, Mourad Chabane Oussalah, Dalila Tamzalit. Une approche basée SOA pour l'interconnexion de workflows : application au "transfert de cas ". 29ème Congrès INFORSID, May 2012, Lille, France. pp.43-58. hal-01064685

HAL Id: hal-01064685

<https://hal.science/hal-01064685v1>

Submitted on 22 Sep 2014

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.

Une approche basée SOA pour l'interconnexion de workflows : application au «transfert de cas »

Saida BOUKHEDOUMA^(*), Zaia ALIMAZIGHI^(*),
Mourad OUSSALAH^(**), Dalila TAMZALIT^(**)

(*) USTHB- FEI- Département Informatique- Laboratoire LSI – Equipe ISI
✉ : El Alia BP n°32, Bab Ezzouar, Alger, Algérie
boukhedouma@yahoo.fr; alimazighi@wissal.dz ;

(**) Université de Nantes- Laboratoire LINA- Equipe MODAL
✉ : 2, Rue de la Houssinière, BP 92208, 44322 Nantes
{Mourad.oussalah, Dalila.tamzalit} @univ-nantes.fr

RESUME. Le workflow et les architectures orientées services (SOA) favorisent la collaboration entre partenaires métiers et aident au développement d'applications collaboratives flexibles, évolutives et efficaces. L'objectif de cet article est de proposer une approche basée sur les services pour l'interconnexion de WF obéissant au schéma de coopération « transfert de cas » dans le but d'obtenir des modèles de WF suffisamment flexibles permettant leur adaptabilité et leur évolutivité. Le « transfert de cas » définit une forme de coopération du WFIO, particulièrement répandue dans les relations B2B regroupant un ensemble de partenaires ayant des objectifs communs, exerçant le même métier et devant satisfaire un grand nombre de clients potentiels. Notre approche repose conceptuellement, sur un méta-modèle de processus combinant les concepts de workflow et de SOA pour la modélisation de processus inter-organisationnels répondant particulièrement, à la forme de coopération « transfert de cas ».

ABSTRACT. Workflow and Service Oriented Architectures (SOA) favor collaboration among business partners and promote the development of flexible and efficient collaborative applications. This paper proposes an approach to connect workflows of several partners using services for modeling inter-organizational processes particularly built according to a "case transfer" form of cooperation. The "case transfer" defines a form of cooperation in IOWF, particularly prevalent in B2B relationships involving a range of partners with common business goals, exercising the same business and aiming to satisfy many potential customers. The advantage of using services is to obtain process models flexible enough in order to allow easier adaptation in case of new business needs, because services are loosely coupled components. The approach is supported by a process meta-model which combines workflow concepts and SOA concepts.

MOTS CLES : processus inter-organisationnel, workflow, service web, SOA, méta-modèle de processus.

KEY WORDS : interorganizational process, workflow, web service, SOA, process meta-model.

1. Introduction

Depuis plusieurs années, la technologie workflow a été largement utilisée dans le milieu des entreprises, pour l'automatisation des processus métiers (Levan, 2000), (Aalst, 2002). Grâce à leurs bénéfices remarquables, les systèmes workflow ont conduit à une amélioration considérable des processus d'entreprises. Cependant, face à la concurrence et dans le but d'améliorer leur productivité, les entreprises expriment un grand besoin d'ouverture et de coopération à l'échelle mondiale. Cette *coopération* interentreprises a été initialement supportée par les outils de workflow *inter-organisationnels (WFIO)* (Aalst, 2000), un concept qui trouve ses origines dans le domaine du e-commerce souvent désigné sous le nom de B2B.

Dans nos travaux de recherche, nous nous intéressons à la *coopération structurée* supportée par le concept de WFIO, mettant en coopération plusieurs WF rattachés à des partenaires métiers afin d'atteindre un objectif commun, selon une politique « gagnant-gagnant ». Dans une coopération structurée, les étapes du processus inter-organisationnel et les points d'interaction entre les différents WF sont bien définis et les instances de processus s'exécutent selon le modèle de WFIO implémenté. Dans la littérature, plusieurs *formes génériques de coopération* ont été définies afin de supporter la coopération structurée devant obéir à un schéma clairement défini. Il s'agit du *partage de charge*, *l'exécution chaînée*, la *sous-traitance*, le *transfert de cas*, le *transfert de cas étendu* et le *workflow faiblement couplé* (Aalst, 1999), (Aalst, 2000). A notre sens, ces formes génériques de coopération résument les différents schémas qui peuvent lier des partenaires dans une relation B2B.

Par ailleurs, dans un contexte de globalisation et de mondialisation, les entreprises se trouvent de plus en plus confrontées à de nombreux défis posés par une concurrence accrue du marché d'une part et par les exigences des clients potentiels d'autre part. Ainsi, ces entreprises sont amenées à améliorer continuellement ou occasionnellement leurs processus métiers et à faire face à des situations imprévues telle la rupture d'un contrat avec un partenaire ou le manque de ressources appropriées pour l'exécution des instances de processus ou autre. En d'autres termes, ces entreprises sont contraintes de revoir leurs systèmes de WF, leurs modèles de processus métiers et leur coopération avec les autres partenaires.

Nous nous intéressons à l'aspect « modèle de processus » WFIO, ces modèles que nous voulons *adaptables* et *évolutifs* en vue de prendre en compte les changements imposés par l'environnement. Pour cela, nous procédons en deux étapes : (1) nous nous penchons sur *l'interconnexion des WF* de manière à faciliter l'adaptabilité et l'évolutivité des modèles de WFIO ; (2) Nous nous pencherons ensuite sur les mécanismes d'adaptabilité et d'évolutivité de ces modèles.

Pour l'interconnexion de WF, nous proposons une *approche basée sur les services* puisque ces derniers sont caractérisés par leur couplage faible, leur interoperabilité et leur facilité d'invocation d'une part, et par leur orientation métier

d'autre part. Notons que le schéma d'interconnexion et les contraintes qui y sont rattachées diffèrent d'une forme de coopération à l'autre.

Le présent article décrit une approche d'interconnexion de WF par les services, adaptée au « *transfert de cas* » (Aalst, 2000). Conceptuellement, notre approche est basée sur un méta-modèle de processus supportant la modélisation de processus métiers inter-organisationnels combinant les technologies WF et SOA (Service Oriented Architecture) (Papazoglou *et al*, 2007). Une SOA est dédiée aux applications collaboratives et distribuées basée sur le concept de *service* pouvant être un *service web* (Alonso *et al*, 2004).

Dans notre approche, le processus WF est découpé en *sous-processus* encapsulés chacun dans un *service* (pouvant être un *service web*). Chaque sous-processus est délimité par deux *points de transfert* dénotant le début et la fin du sous-processus et permettant le transfert d'instances, à travers des opérations d'invocation de services.

Le reste de l'article est structuré comme suit : la section 2 situe le contexte du travail et les principaux concepts qui y sont rattachés. La section 3 expose brièvement les travaux similaires et met en évidence la motivation de ce travail. La section 4 présente l'approche proposée. La section 5 décrit un exemple de processus inter-organisationnel implémentant une coopération du type « *transfert de cas* » entre deux partenaires fournisseurs de médicaments à des clients potentiels, avec quelques détails techniques. La section 6 conclut le travail en mettant l'accent sur quelques unes de ses extensions.

2. Contexte du travail

Le WFIO peut être défini comme un gestionnaire d'activités faisant intervenir deux ou plusieurs WF autonomes, interopérables et éventuellement hétérogènes dans le but d'atteindre un objectif métier commun (Aalst, 2000).

Parmi les formes génériques de coopération définies dans la littérature du WFIO, le « *transfert de cas* » est assez répandu dans les échanges B2B, notamment entre partenaires exerçant le même métier et devant satisfaire un grand nombre de clients potentiels, dans les meilleurs délais, d'où l'intérêt de leur alliance. Dans le « *transfert de cas* », les partenaires métiers partagent tous le *même modèle de processus* implémenté au niveau de chacun d'eux. Leur coopération consiste à transférer l'exécution d'une instance de processus d'une localisation (i.e un partenaire) à une autre afin de prendre en charge une partie ou la totalité de son exécution. Les règles de transfert sont décrites dans la *politique de transfert* définie conjointement par les différents partenaires.

A tout moment, une instance de processus se trouve au niveau d'un seul partenaire, le transfert de l'instance doit se faire à partir d'un point stable du processus appelé *point de transfert* et doit prendre en compte les *données d'état* de l'instance afin de maintenir la cohérence du système. Le transfert d'instances se fait

pour des raisons d'équilibrage de charge entre les partenaires ou à cause d'une indisponibilité de ressources adéquates ou autre. La figure 1 schématise une coopération entre trois partenaires métiers hébergeant tous le même modèle *WF1*.

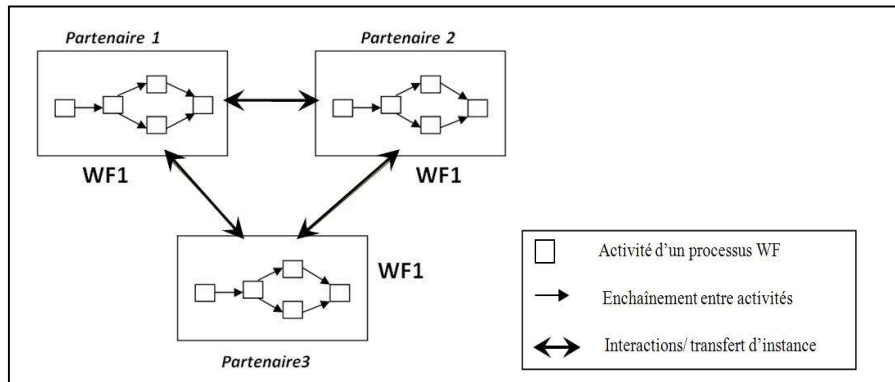


Figure 1. Schéma de WFIO selon le modèle « transfert de cas »

Données d'état d'une instance : les données d'état d'une instance de processus sont principalement: (1) les informations générales sur l'instance, (2) l'historique d'exécution et (3) l'activité à exécuter directement après le transfert de l'instance.

Définition : Point de transfert

Un point de transfert est un état du processus qui garantit la *cohérence* d'exécution de l'instance si un transfert est effectué d'une localisation à une autre.

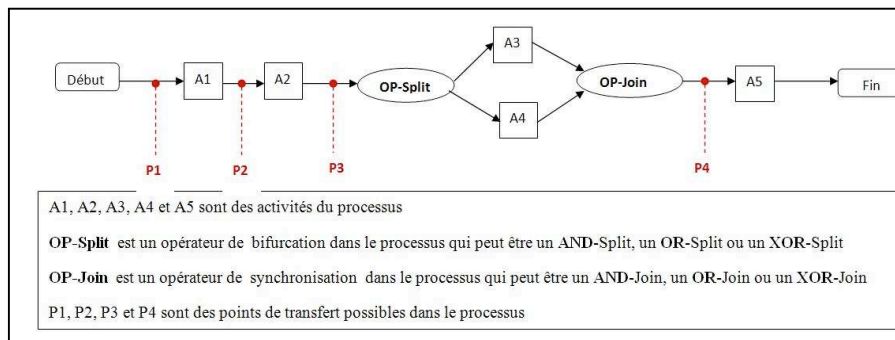


Figure 2. Illustration du point de transfert

En effet, un point de transfert remplit les conditions suivantes : (1) doit se situer avant le début d'une activité ou après la fin d'une activité. (2) Ne doit pas interrompre l'exécution d'une activité. (3) Ne doit pas se situer entre un opérateur de routage (SPLIT) et l'opérateur de synchronisation (JOIN) correspondant. En d'autres termes, si une branche parallèle ou alternative est engagée, le transfert d'instance ne peut s'effectuer qu'après synchronisation.

3. Travaux similaires et motivation

Avec l'émergence des SOA et la technologie des services web, beaucoup de travaux se sont orientés vers le développement d'applications métiers collaboratives autour de services. Ainsi, certains travaux se sont focalisés sur la composition dynamique de services (Casati *et al.*, 2000), (Fensel *et al.*, 2003), (Chen *et al.*, 2005), sur l'orchestration et la chorégraphie de services (Amirreza, 2009), (Peltz, 2003), (Decker *et al.*, 2007) basés essentiellement sur le langage BPEL4WS. D'autres travaux ont axé sur l'intérêt de combiner le BPM (Business Process Management) et les SOA pour la réutilisation de services dans la construction de processus métiers dynamiques (Leymann *et al.*, 2002), (Gorton *et al.*, 2009). Par ailleurs, La plupart des travaux orientés SOA (Lopez-Sanz *et al.*, 2008), (Arsanjani *et al.*, 2008) considèrent des applications métiers collaboratives qui sont complètement automatiques et ne prennent pas en compte l'intervention humaine dans le processus. De plus, ils se focalisent sur les aspects techniques plutôt que sur les aspects conceptuels.

Certains travaux tels que (Perrin *et al.*, 2004), (Belhajiame *et al.*, 2005) et (Chebbi, 2007) se sont orientés vers la proposition d'approches et de plates-formes supportant le WFIO dans le cadre d'une coopération dynamique structurée.

Dans nos travaux de recherche, nous nous intéressons à la coopération structurée dans le domaine du WFIO, basée sur les formes génériques de coopération définies dans (Aalst, 1999), (Aalst, 2000). Celles-ci représentent, à notre sens, des patrons de base pour la coopération de workflows. Notre objectif à moyen terme, est de proposer des approches et des mécanismes pour réaliser l'adaptabilité et l'évolutivité des modèles de processus dans ce contexte. Notre motivation découle du fait que les entreprises sont souvent confrontées à de nouveaux défis qui les obligent à revoir leurs systèmes de WF, en particulier leurs modèles de processus.

Afin de garantir un degré de flexibilité élevé des modèles de WFIO, nous adoptons une approche d'interconnexion des WF par les *services* vues leurs caractéristiques : faiblement couplés, interopérables, orientés métiers et invocables.

Dans cet article, nous nous focalisons sur la forme de coopération du type *transfert de cas*. Nous proposons un schéma d'interconnexion de WF en utilisant les *services*, supporté conceptuellement par un méta-modèle de processus inter-organisationnel combinant les concepts de workflow et de SOA.

4. Notre approche

L'interconnexion entre les WF des différents partenaires est réalisée via les services. En effet, les interactions (les opérations de transfert d'instances) entre les WF hébergés au niveau des différents partenaires se font à travers des opérations d'*invocation* comme le montre la figure 4. Ainsi, deux questions élémentaires se posent : où seront situés les services par rapport au workflow ? Comment se fait le contrôle d'exécution du processus lors du transfert d'instances ?

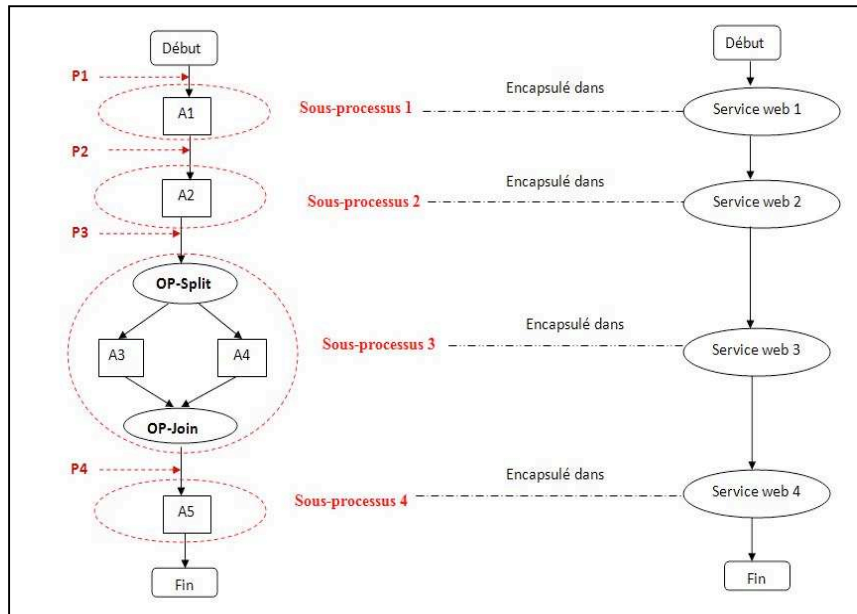


Figure 3. Correspondance entre sous-processus et services

Pour répondre à la première question, nous proposons de *partitionner* le processus WF en *sous-processus* délimités chacun par deux *points de transfert* dénotant respectivement, le début et la fin du sous-processus afin de garantir la cohérence d'exécution de l'instance lors du transfert. Ainsi, chaque *sous-processus est encapsulé dans un service*, particulièrement un *service web* (voir figure 3) permettant son invocation et possède deux activités principales : une activité d'entrée (Input) permettant son invocation lors du transfert et une activité de sortie (Output) permettant le renvoi de résultats fournis par le service.

Remarque : le partitionnement du processus en sous-processus dépend des *points de transfert* choisis par les concepteurs du WFIO, selon la politique de transfert adoptée par les partenaires. Ainsi, un même service peut encapsuler une ou plusieurs activités reliées entre elles par des opérateurs de contrôle de flux.

Pour répondre à la deuxième question (contrôle d'exécution), deux architectures sont possibles: une architecture *avec coordinateur central* qui gère les interactions entre les différents WF ou une architecture *sans coordinateur central*. Le choix de l'architecture avec ou sans coordinateur central dépend de la complexité de la politique de transfert et du choix des partenaires.

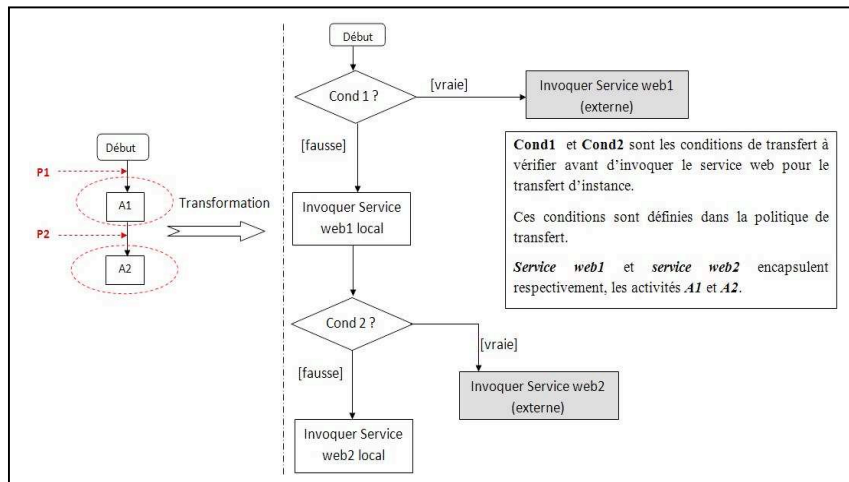


Figure 4. Transformation du modèle de processus

Dans ce travail, nous considérons une architecture *sans coordinateur central*. Dans ce cas, les opérations d’invocation de services pour le transfert d’instances entre deux WF se font directement à partir des moteurs de WF interprétant les modèles de processus au niveau de chaque partenaire. Par suite, chaque WF demandeur de transfert doit comporter une activité particulière du type « *invoker service* » au niveau de chaque point de transfert (voir figure 4), véhiculant les données d’état de l’instance et les artefacts nécessaires à son exécution. Les invocations de services se font sous les *conditions* définies dans la *politique de transfert*. La figure 4 montre la transformation d’une partie d’un modèle de processus WF (comportant deux activités séquentielles A1 et A2) après insertion des activités d’invocation. Si la condition (cond1 ou cond2) de transfert est vraie, l’instance est transférée à un autre partenaire via l’invocation du service (externe) à exécuter ; dans le cas contraire, le service est exécuté localement.

Dans la suite, nous décrivons le méta-modèle de processus global, qui est le support conceptuel de notre approche.

4.1 Méta-modèle de WFIO pour l’architecture transfert de cas

Le méta-modèle de processus décrit dans la figure 5 sous forme d’un diagramme de classes UML est inspiré du méta-modèle de définition de WF proposé par la WFMC. Nous l’avons étendu aux concepts permettant de prendre en compte l’aspect inter-organisationnel et quelques concepts du paradigme SOA afin de supporter notre approche d’interconnexion de WF, conformément au schéma *transfert de cas*.

Par ailleurs, l’approche par méta-modèle garantit la conformité des modèles construits par rapport aux concepts identifiés et facilite l’adaptation de modèles en cas de nouveaux besoins métiers. Ainsi, ce méta-modèle de processus sert de cadre à la construction de modèles de processus conformément aux concepts identifiés.

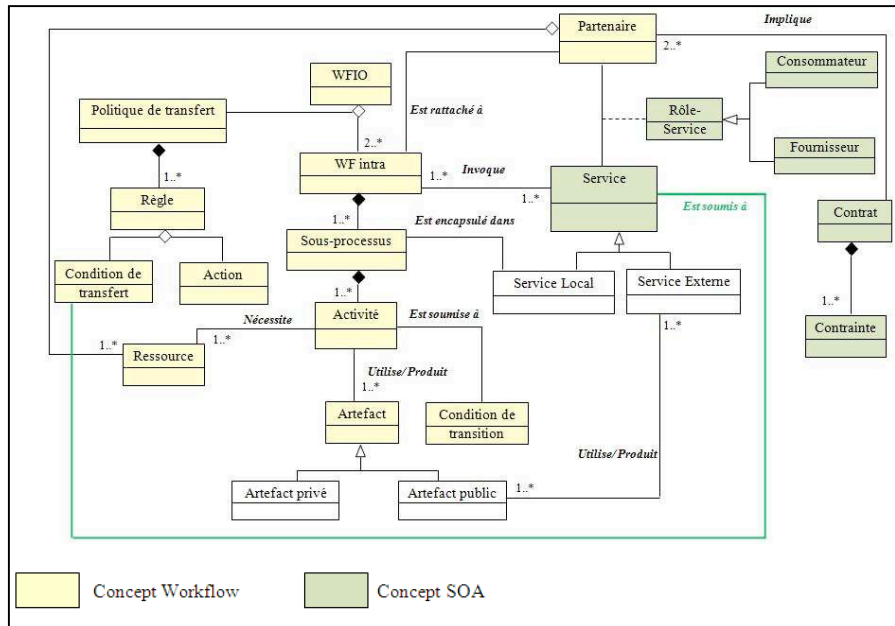


Figure 5. Méta-modèle de processus WFIO selon la forme de coopération « transfert de cas »

Un WFIO est composé d'un ensemble de WF *intra-organisationnels* identiques et d'une *politique de transfert* définie conjointement par l'ensemble des partenaires impliqués dans la coopération. La politique de transfert est un ensemble de *règles de transfert*, elle définit *quand, vers où et sous quelles conditions* faut-il transférer une instance de processus en cours d'exécution. La politique de transfert contrôle ainsi, les invocations de services entre les workflows impliqués dans le processus global.

La WFCM et certains travaux liés à la modélisation workflow (Saikali, 2001) identifient des aspects de modélisation complémentaires de workflow; le méta-modèle de la figure 5 couvre quatre aspects de modélisation de processus que nous décrivons dans les sections suivantes.

4.1 Aspect processus

Cet aspect de modélisation décrit d'une part, la décomposition du WFIO en deux ou plusieurs WF *intra-organisationnels* (WF intra) et d'une *politique de transfert* (voir figure 6). Chaque WF intra-organisationnel est composé d'un ensemble de *sous-processus*; chaque sous-processus est composé d'*activités* et est complètement encapsulé dans un *service local* (implémenté localement). Un WF intra-organisationnel invoque un ou plusieurs services. Le service invoqué peut être *local* si l'instance ne nécessite pas de transfert, il peut être *externe* si l'instance nécessite un transfert. Un service local à un WF externe à un autre WF et vice versa. L'invocation d'un service est soumise à une *condition de transfert*. En effet, si

la condition de transfert est vraie, c'est le service local adéquat qui est invoqué sinon c'est le service externe qui est invoqué.

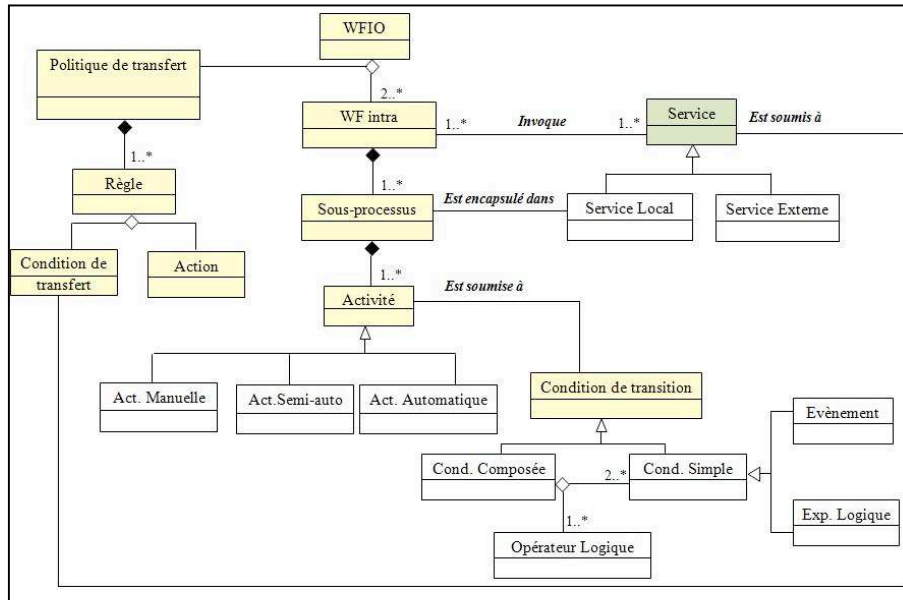


Figure 6. Méta-modèle fonctionnel et comportemental (Aspect processus)

L'aspect processus décrit d'autre part, le contrôle de flux des activités au niveau d'un sous-processus, à savoir les points de disjonction, conjonction, synchronisation imposés par les conditions de transitions. L'enchaînement des activités du sous-processus est exprimé à travers des opérateurs appropriés de contrôle de flux. Une condition peut être simple ou composée, une condition simple est soit une expression logique sur les données du WF soit un événement. Une activité peut se spécialiser en activité manuelle, semi-automatique ou automatique.

4.2 Aspect organisationnel

L'aspect organisationnel met en évidence les participants impliqués dans le déroulement du processus. Il fait ressortir les partenaires impliqués dans la coopération, le fournisseur de service et le consommateur de service. Pour un service donné, un partenaire peut jouer le rôle de fournisseur et/ou de consommateur de ce service exprimé via le concept *Rôle-Service* (voir figure 7). Ceci obéit particulièrement à une architecture SOA avec les composants essentiels de fournisseur de service, consommateur de service et le service lui-même. Par ailleurs au niveau interne, chaque partenaire dispose d'un ensemble de ressources pouvant se spécialiser en ressource humaine (tenant un ou plusieurs rôles), matérielle ou logicielle chargées de l'exécution des différents types d'activités.

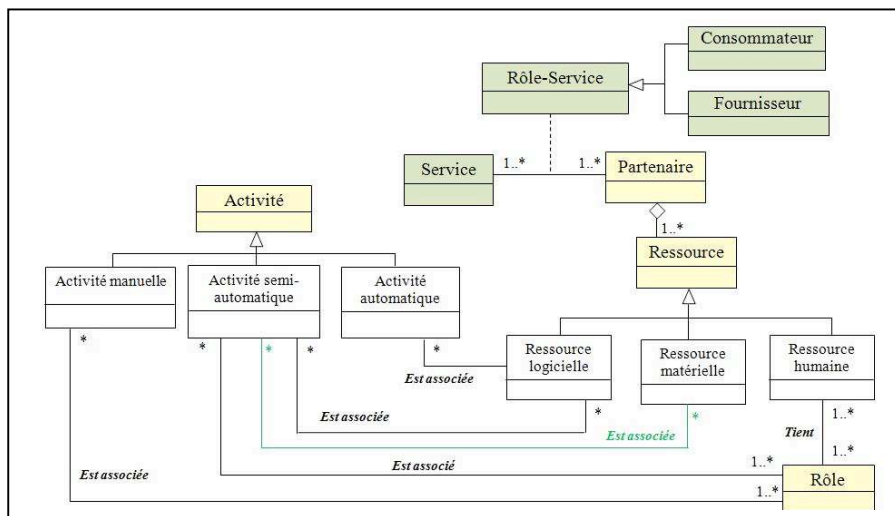


Figure 7. Méta-modèle organisationnel

4.3 Aspect interactionnel

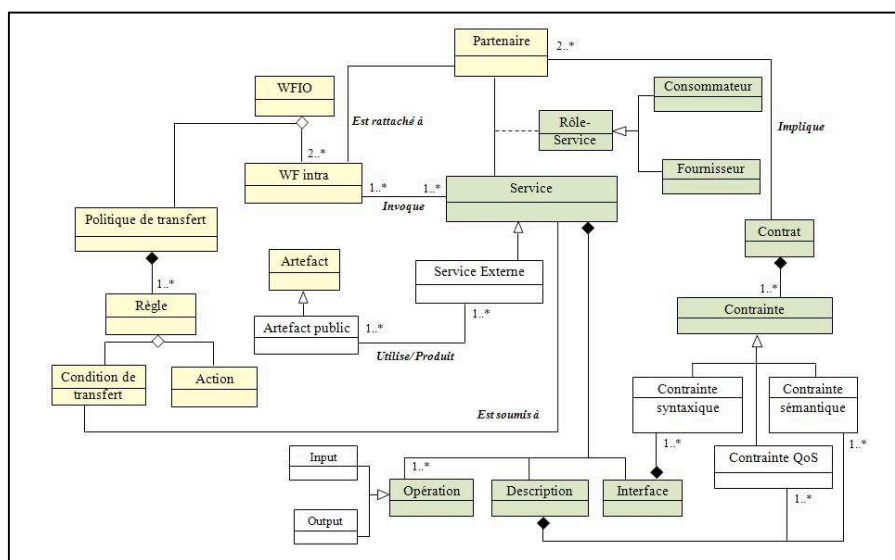


Figure 8. Méta-modèle interactionnel

La communication entre le WF initiateur du transfert et le WF destinataire se fait via l'invocation d'un *service externe* (voir figure 8), seules les opérations d'entrée (input) et de sortie (output) de données échangées sont visibles de l'extérieur.

L'interface et la description du service sont les éléments qui assurent la *visibilité* du service. *L'interface* fournit les informations nécessaires et suffisantes à une communication avec le service mais ne fournit pas de garantie sur la bonne utilisation de ce service. Elle regroupe les contraintes *syntaxiques*. *La description de service* contient des informations complémentaires à *l'interface* qu'elle décrit pour permettre une *interaction* correcte. Elle contient les contraintes *sémantiques* et de *QoS (qualité de service)*. Un *service externe* utilise et/ou produit des *artefacts publics* qui sont typiquement les *données d'état* de l'instance et les *artefacts* avant et après l'invocation du service. Les transferts d'instances de processus se font conformément à la politique de transfert définie, ainsi un transfert d'instance se fait selon une règle de transfert définie par la paire (condition, action) signifiant si la condition de transfert est vraie alors exécuter l'action correspondante qui définit essentiellement vers où il faut transférer l'instance.

4.4 Aspect informationnel

L'aspect informationnel est supporté par le concept générique d'*artefact* pouvant se spécialiser en une donnée, un fichier, un formulaire, etc. En effet, un artefact représente toute information nécessaire à l'exécution d'une activité (donc du sous-processus qui la contient) ou produite par une activité. Certains artefacts sont *publics* et d'autres sont *privés* (voir figure 5). Un artefact *public* peut être vu et manipulé par les *services externes*. Dans notre cas, il s'agit en effet, de toute information véhiculée par les messages lors des invocations de services externes pour le transfert d'instances de processus, donc les *données d'état* de l'instance transférée et les *artefacts* nécessaires à l'exécution de l'instance. Un *artefact privé* par contre, n'est visible qu'à l'intérieur de l'organisation et ne peut être manipulé que par les activités implémentées localement, donc les services locaux.

5. Illustration de l'approche

Afin d'illustrer l'approche proposée, nous décrivons un processus inter-organisationnel impliquant deux organismes producteurs et fournisseurs de médicaments devant satisfaire un grand nombre de clients potentiels. Le processus en question concerne la commercialisation de médicaments à des clients potentiels et peut être implémenté selon la forme de coopération « *transfert de cas* » avec une politique de transfert très simple. Le processus que nous décrivons est largement inspiré d'une étude de cas réelle mais que nous avons dû adapter pour des besoins d'illustration de l'approche. De ce fait, nous attribuons des noms fictifs aux organismes impliqués dans le processus, respectivement PHARMA et MEDIC. Les commandes émanant des clients arrivent au niveau des deux organismes et sont transférées d'un partenaire à l'autre si la licence de production n'est pas disponible au niveau du partenaire qui reçoit la commande.

5.1 Description du processus

Le diagramme d'activité de la figure 9 décrit globalement, le processus inter-organisationnel reliant les deux organismes PHARMA et MEDIC, à travers l'invocation d'un *service web externe* « Traiter BCT » (bon de commande transféré).

Le client fait une commande via le site web d'un partenaire; deux cas sont possibles : (1) la commande concerne des médicaments produits localement, elle est traitée localement à travers l'invocation du *service local* « Traiter BC ». (2) La commande concerne des médicaments produits par le partenaire externe, elle est aussitôt transférée pour traitement au niveau de ce partenaire, via le *service externe* « Traiter BCT ». Le processus repose sur une politique de transfert très simple puisqu'il comporte un seul point de transfert.

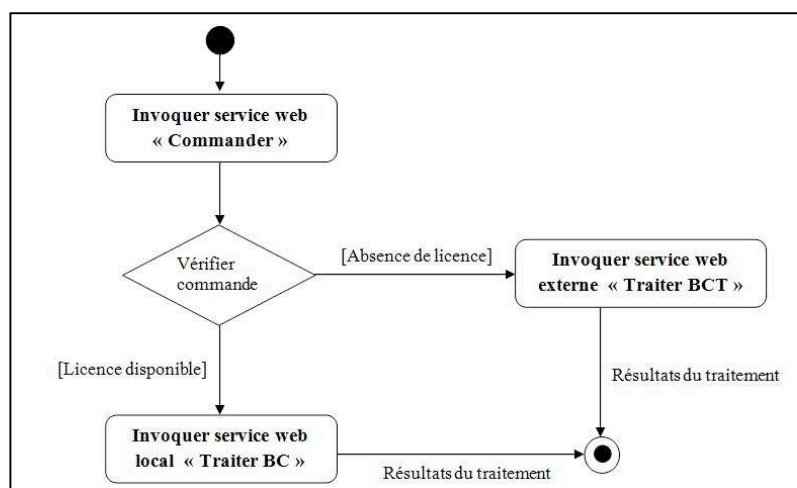


Figure 9. WFIO « Gérer commandes PHARMA/MEDIC »

Les services « Traiter BC » et « Traiter BCT » encapsulent le même sous-processus implémenté au niveau des deux organismes. La figure 10 donne une description de ce sous-processus qui s'interprète comme suit :

- (1) L'agent de vente approuve le BP (Bon de Préparation) et l'envoie au service préparation.
- (2) Le préparateur prépare la marchandise qui sera contrôlée au service contrôle.
- (3) Si la marchandise est conforme, un bon de livraison (BL) est établi et envoyé au validateur et à l'agent d'expédition. Sinon, un rapport de non-conformité est établi et envoyé au préparateur.
- (4) Le validateur du service contrôle valide le BL, prépare une facture et l'envoie au service comptabilité et à l'agent d'expédition.
- (5) Le validateur envoie un email de notification au client pour l'informer que sa commande est prête.
- (6) Le comptable suit le paiement de la facture et envoie un bon d'expédition (BE).
- (7) L'agent d'expédition livre la marchandise au client.

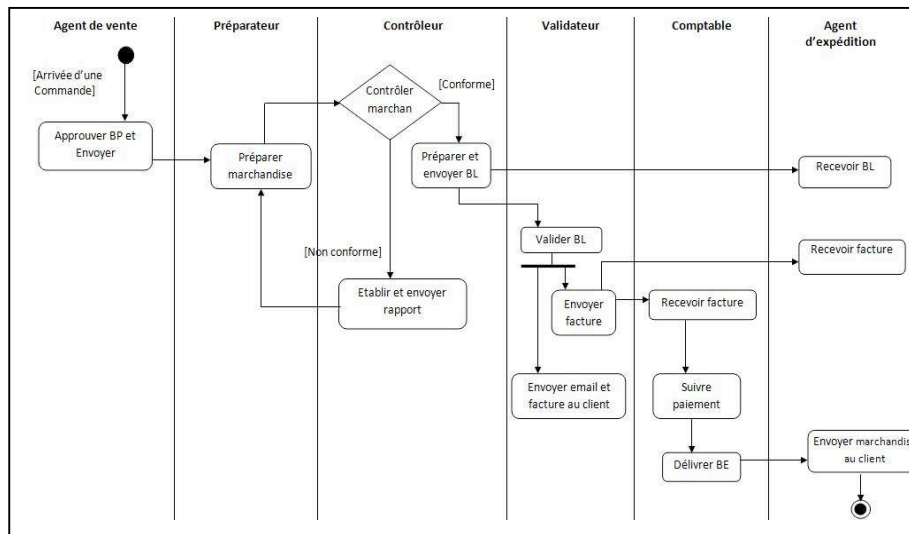


Figure 10. Diagramme d'activité du sous-processus « traiter BC (BCT) »

Pour cet exemple, le concept de *politique de transfert* est décrit par une seule règle notée Règle1 pouvant être exprimée comme suit :

Règle 1 : si les médicaments commandés ne sont pas disponibles localement (absence de la licence de production) alors transférer l'instance à l'autre partenaire.

5.2 Correspondance des concepts avec le langage jPDL

Afin d'implémenter le processus précédemment conçu, nous avons choisi l'environnement jBOSS/jBPM qui constitue un cadre flexible et extensible pour l'exécution des processus métiers permettant de coordonner entre acteurs, applications et services. jPDL est un des langages de spécification de processus qui se construit au-dessus de ce cadre commun. Il exprime graphiquement les processus métiers en termes de tâches, d'états d'attente pour la communication asynchrone et d'autres composants. Ce langage est interprétable par le moteur de workflow jBPM qui s'exécute sous jBOSS permettant, en combinaison avec ESB (Entreprise Service Bus), l'invocation de services en utilisant des classes java. L'intégration jBPM/ESB permet d'une part d'invoquer des services à partir d'un processus métier et d'autre part d'exposer un processus métier tel un service ou un ensemble de services. Le tableau 1 établit une correspondance de concepts entre le méta-modèle de processus et jPDL. Le concept de « *process-state* » dans jPDL correspond à l'invocation d'un sous-processus à partir d'un processus parent (processus invocateur). L'instance du processus « parent » s'arrête pendant tout le temps d'exécution du sous-processus invoqué et reprend l'exécution lorsque ce dernier est terminé.

Concept du méta-modèle de processus	Concept jPDL
Processus/ sous-processus	process/sub-process
Activité manuelle / semi-automatique	task-node
Activité automatique	Node
Rôle/ Ressource humaine	Swimlane/ Actor
Artefact	Variable
Enchaînement séquentiel	Transition
Opérateur de parallélisme/synchronisation	Fork/join
Opérateur de disjonction	Decision

Tableau 1. Correspondance de concepts entre le méta-modèle et jPDL

5.3 Architecture de la solution

Rappelons que nous avons considéré une architecture sans coordinateur central. Ainsi, les règles de transfert sont injectées dans la spécification du processus sous la forme (condition, action). Après évaluation de la condition de transfert par le SGWF (système de gestion de workflow) local, si elle est vraie, le SGWF invoque *directement* le service externe approprié via le SGWF du partenaire adéquat.

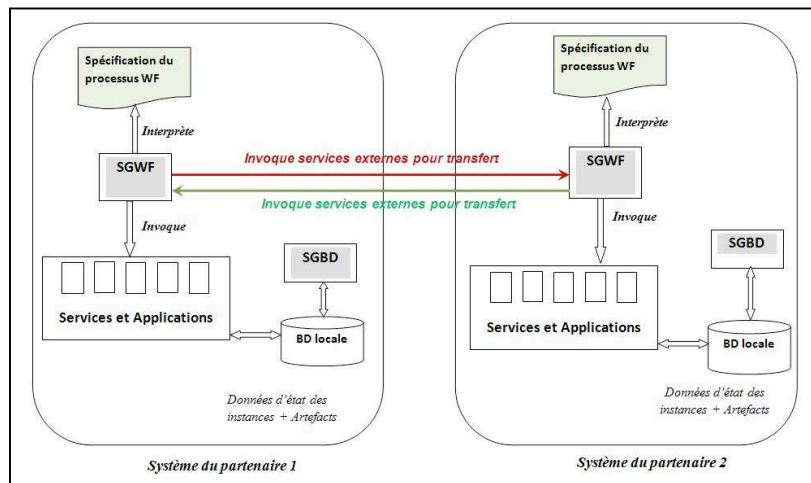


Figure 11. Architecture du système

L'opération d'invoque transmet les données d'état de l'instance ainsi que les artefacts nécessaires à son exécution. Lorsqu'une instance est transférée, le SGWF qui la reçoit gère la suite de son exécution conformément à la spécification du processus implémenté localement en utilisant les applications et les services locaux. Cette architecture s'avère appropriée dans le cas d'une politique de transfert simple (càd les règles de transfert sont déterministes)

6. Conclusion

Dans nos travaux de recherche, nous nous intéressons à la coopération structurée dans le domaine du WFIO, nous nous basons notamment sur les *formes génériques de coopération* définies dans (Aalst, 1999), (Aalst, 2000), considérées comme des patrons de base du WFIO. Notre objectif final est de réaliser l'adaptabilité et l'évolutivité des modèles de WFIO, de ce fait nous adoptons une approche orientée *services* pour l'interconnexion de WF permettant une modification aisée des modèles. Notons que le schéma d'interconnexion et les contraintes qui y sont rattachées diffèrent d'une forme de coopération à l'autre.

Dans nos travaux récents, nous avons considéré une coopération du type *sous-traitance* (Boukhedouma et al, 2010a), (Boukhedouma et al, 2010b). Dans cet article, nous nous sommes focalisés sur le *transfert de cas* (Aalst, 1999). Il s'agit d'un schéma de coopération assez répandu entre partenaires métiers exerçant le même métier et devant répondre aux exigences d'un nombre important de clients potentiels. Le *transfert de cas* suppose un partitionnement vertical du modèle de workflow, puisque le modèle de processus est dupliqué au niveau de chaque partenaire et la coopération consiste à transférer les instances d'une localisation à une autre pour exécution. Le transfert d'instances se fait selon une politique de transfert définie par les différents partenaires impliqués dans la coopération.

Nous avons proposé une approche d'interconnexion de WF basée sur les services et supportée conceptuellement par un méta-modèle de processus combinant les concepts du WF et ceux du paradigme SOA. L'idée principale de notre approche est de partitionner correctement le modèle de processus commun aux différents partenaires, en *sous-processus* afin d'encapsuler chacun d'eux dans un service pouvant être invoqué pour le transfert d'instances. Un sous-processus est encadré par deux points de transfert, une notion que nous avons définie dans cet article.

Dans cet article, nous avons considéré une architecture sans coordinateur central. Comme perspectives à ce travail, nous envisageons de considérer une architecture avec coordinateur central et d'établir une comparaison entre les deux approches.

Actuellement, nous nous intéressons à d'autres formes de coopération à savoir l'*exécution chaînée*, et le WFIO *faiblement couplé*.

7. Bibliographie

- (Aalst, 1999): Aalst W.V.D., "Process-oriented architectures for electronic commerce and interorganizational workflow", *Journal of Information systems, Elsevier Sciences*. 1999.
- (Aalst, 2000): Aalst W.V.D., "Loosely Coupled Interorganizational Workflows : modeling and analyzing workflows crossing organizational boundaries", *Journal of Information and Management*, Vol. 37, p. 67-75 Issue 2, March 2000.
- (Aalst, 2002): Aalst W.V.D., *Workflow Management: Models, Methods and Systems*, The MIT Press, Cambridge, Massachusetts, London, England, 2002.
- (Alonso et al, 2004): Alonso G., Casati F., Kuno H., "Web services: concepts, architectures and applications", *Springer Verlag*, Heidelberg, Germany, 2004.

- (Amirreza, 2009)** : Amirreza T., *Web Service Composition Based Interorganizational Workflows*, Sudwestdeutscher Verlag für Hochschulschriften edition, 2009.
- (Arsanjani et al, 2008)**: Arsanjani A., Ghosh S., Allam A., Abdollah T., Ganapathy S., and Holley K., “A Method for Developing Service Oriented Solutions”. *IBM Systems journal*, 47(3): 377-396. 2008
- (Belhajjame et al, 2005)**: K. Belhajjame, G. Vargas-Solar, and C. Collet. “Pyros - an environment for building and orchestrating open services”. In *SCC '05 : Proceedings of the 2005 IEEE International Conference on Services Computing*, pages 155–164, Washington, DC, USA, 2005. *IEEE Computer Society*.
- (Boukhedouma et al, 2010a)** : Boukhedouma S., Alimazighi Z., “A process meta-model based method for the development of collaborative applications built on workflow and SOA”, in *proceedings of EMCIS'2010- UAE* – 12, 13 Avril, 2010.
- (Boukhedouma et al, 2010b)** : Boukhedouma S., Alimazighi Z., « Un Méta-modèle de processus métiers inter-organisationnels : une approche basée workflow et SOA », *ateliers SIRE'2010 associé à INFORSID'2010*, Marseille, Mai 2010.
- (Casati et al, 2000)**: Casati F., Ilnicki S., Jin L., Krishnamoorthy V., Shan M.C., “Adaptive and Dynamic Service Composition in eFlow”, *HP- Laboratories. HLP-2000-39*, 2000.
- (Chebbi, 2007)** : Chebbi I., CoopFlow : une approche pour la coopération ascendante de workflows dans le cadre des entreprises virtuelles, Thèse de doctorat de l'Institut National des Télécoms, France, 2007.
- (Chen et al, 2005)**: Chen M., Zhang D., Zhou L., “Empowering collaborative commerce with web services enabled business process management system, Decision Support System”, 2005. www.sciencedirect.com
- (Decker et al, 2007)**: Decker G., Kopp O., Leymann F., and Weske M., “Bpel4chor:Extending bpel for modeling choreographies”. In *Proceedings of the International Conference on Web Services (ICWS)*, IEEE Computer Society, 2007.
- (Fensel et al, 2003)**: Fensel D., Bussler C., “The Web Service Modeling Framework WSMF”, www.wsmo.org/papers/publications/wsmf, University of Amsterdam, 2003.
- (Gorton et al, 2009)**: Gorton S., Montangero C., Reiff-Marganiec S., Semini L., “StPowla: SOA, Policies and Workflows”, *ICSOC 2007 workshops, LNCS 4907*, p. 351-362, 2009.
- (Leymann et al, 2002)**: Leymann, F., Roller D., and Schmidt M.T., “Web Services and Business Process Management”, *IBM Systems Journal*, vol. 41, No. 2, 2002.
- (Levan, 2000)**: Levan S.K., *Le projet workflow- concepts et outils au service des organisations*, 2ème tirage, Eyrolles, Avril 2000.
- (Lopez-Sanz et al, 2008)** Lopez-Sanz M., Acuna C.J., Cuesta C.E., and Marcos E., “Modelling of service oriented architectures with UML”, *Electronics Notes in Theoretical Computer Science*, 194: 23-37. 2008
- (Papazoglou et al, 2007)**: Papazoglou M. P., Heuvel W., “Service Oriented Architectures : approaches, technologies and research issues”, *the VLDB Journal*, vol. 16, p. 389-415, Springer Verlag, March 2007.
- (Peltz, 2003)**: Peltz, C., “Web Services Orchestration and Choreography”, *IEEE Computer*, vol. 36, No. 10:46-52, October 2003.
- (Perrin et al, 2004)**: O. Perrin and C. Godart. “A model to support collaborative work in virtual enterprises”. *DataKnowledge Engineering*, 50(1) :63–86, 2004.
- (Saikali, 2001)** : Saikali, K., Flexibilité des workflows par l'approche objet : 2flows, un framework pour les workflows flexibles. Thèse de doctorat de l'école centrale de Lyon, France. 2001.