

# Les Web services sémantiques

Patrick Kellert et Farouk Toumani

Laboratoire LIMOS - UMR (6158) du CNRS  
ISIMA - Campus des Cezeaux - B.P. 125  
63173 AUBIERE Cedex  
<http://www.isima.fr/limos/>  
{patrick.kellert,ftoumani}@isima.fr

## **Résumé**

*Les Web services sémantiques se situent à la convergence de deux domaines de recherche importants qui concernent les technologies de l'Internet, à savoir le Web sémantique et les Web services. Ce chapitre s'efforce de définir le concept de Web service sémantique et son apport potentiel aux technologies existantes (Web services et workflow). Un recensement des problématiques de recherche actuelles est proposé et des directions pour les recherches futures dans ce domaine sont suggérées.*

**Mots-clés :** *Web services, Web sémantique, Ontologies, QoS, E-Workflow.*

## **Abstract**

*Semantic Web services are at the convergence of two research fields related to Internet technologies, namely semantic Web and Web services. This chapter aims at defining the concept of semantic Web service and its potential contribution to existing technologies (Web services and workflow). A census of current research problems is proposed and future research directions in this field are highlighted.*

**Keywords :** *Web services, Semantic Web, Ontologies, QoS, E-Workflow.*

# 1 PRESENTATION ET IMPORTANCE DE LA PROBLEMATIQUE DU POINT DE VUE DES USAGES

Les *Web services sémantiques* [13, 15, 34] se situent à la convergence de deux domaines de recherche importants qui concernent les technologies de l'Internet : le Web sémantique et les *Web services*. Le Web sémantique s'intéresse principalement aux informations statiques disponibles sur le Web et les moyens de les décrire de manière intelligible pour les machines (cf. l'introduction et le chapitre 2 sur les langages). Les *Web services*, quant à eux, ont pour préoccupation première l'interopérabilité entre applications via le Web en vue de rendre le Web plus dynamique.

La notion de «Web service» désigne essentiellement une application (un programme) mise à disposition sur Internet par un fournisseur de service, et accessible par les clients à travers des protocoles Internet standards [15, 9]. Des exemples de services actuellement disponibles concernent les prévisions météorologiques, la réservation de voyage en ligne, les services bancaires ou des fonctions entières d'une entreprise comme la mise en œuvre de la gestion de la chaîne logistique.

Le consortium W3C <<http://www.w3.org/2002/ws/>> définit un Web service comme étant une application ou un composant logiciel qui vérifie les propriétés suivantes :

- Il est identifié par un URI ;
- Ses interfaces et ses liens (*binding*) peuvent être décrits en XML ;
- Sa définition peut être découverte par d'autres Web services ;
- Il peut interagir directement avec d'autres Web services à travers le langage XML et en utilisant des protocoles Internet.

L'objectif ultime de l'approche Web services est de transformer le Web en un dispositif distribué de calcul où les programmes (services) peuvent interagir de manière intelligente en étant capables de se découvrir automatiquement, de négocier entre eux et de se composer en des services plus complexes [15, 34]. En d'autres termes, l'idée poursuivie avec les Web services, est de mieux exploiter les technologies de l'Internet en substituant, autant que possible, les humains qui réalisent actuellement un certain nombre de services (ou tâches), par des machines en vue de permettre une découverte et/ou une composition automatique de services sur l'Internet. L'automatisation est donc un concept clé qui doit être présent à chaque étape du processus de conception et de mise en

œuvre des Web services. Comme mentionné dans [9, 15], l'automatisation est essentielle pour intégrer les facteurs suivants :

- Passage à l'échelle : il faut être capable de traiter un nombre important de Web services (annuaire de services au niveau mondial).
- Forte réactivité dans un environnement hautement dynamique.
- Réduction des coûts de développement et de maintenance des Web services.

On peut de plus rajouter les facteurs suivants :

- Forte adaptabilité facilitant la maintenance et l'évolution des Web services : il est vraisemblable que vu, l'enjeu que représente leur réussite et de par leur orientation métier, les Web services créés seront amenés à être modifiés fréquemment.
- Prise en compte de critères de qualité de services aussi bien d'un point de vue qualitatif que quantitatif : il est clair que la plupart des critères de qualité de services proposés actuellement (*e.g.*, le prix) ne prennent pas en compte des aspects qualitatifs (*e.g.*, la notion de réputation d'un fournisseur [32]).

Or la plupart des travaux existants qui s'intéressent à l'intégration fonctionnelle évitent le problème fondamental de l'automatisation des différentes étapes liées à la fourniture d'un Web service (par exemples, découverte et composition) puisqu'ils limitent l'usage des Web services aux utilisateurs humains plutôt qu'aux machines. En effet, de nombreuses connaissances, indispensables pour l'automatisation des services, sont soit absentes, soit décrites pour être interprétées et exploitées par des humains. Il en résulte un rôle prédominant pour le programmeur humain. Il semble donc nécessaire de tendre vers des services intelligibles pour des machines : c'est le concept de Web service sémantique.

Le besoin d'automatisation du processus de conception et de mise en œuvre des Web services rejoint les préoccupations à l'origine du Web sémantique, à savoir comment décrire formellement les connaissances de manière à les rendre exploitables par des machines. En conséquence, les technologies et les outils développés dans le contexte du Web sémantique peuvent certainement compléter la technologie des Web services en vue d'apporter des réponses crédibles au problème de l'automatisation. Par exemple, la notion d'ontologie peut jouer un rôle prépondérant pour permettre d'explicitier la sémantique des services facilitant ainsi les communications hommes-machines, d'une part, et les communications machines-machines, d'autre part.

De manière générale, l'objectif visé par la notion de Web services sémantiques est de créer un Web sémantique de services dont les propriétés, les capacités, les interfaces et les effets sont décrits de manière non ambiguë et exploitable par des machines et ce en utilisant les couches techniques sans pour autant en être conceptuellement dépendants. La sémantique ainsi exprimée permettra l'automatisation des fonctionnalités suivantes qui sont nécessaires pour une collaboration inter-entreprises efficace :

- processus de description et de publication des services ;
- découverte des services ;
- sélection des services ;
- composition des services ;
- fourniture et administration des services ;
- négociation des contrats.

## **2 METHODES, TECHNIQUES, OUTILS EXISTANTS SUR LESQUELS ON PEUT S'APPUYER**

Les Web services tendent à devenir un domaine de recherche à part entière qui suscite beaucoup d'intérêt de la part de chercheurs de communautés très variées. On peut citer à titre d'exemple, le génie logiciel, les *workflows*, les bases de données, la modélisation d'entreprises, la représentation des connaissances ou les multi-agents. Cependant, on constate aujourd'hui que la littérature scientifique traitant des Web services est trop dispersée. Il en résulte une absence d'unification et d'intégration de ses concepts rendant, tout au moins actuellement, difficile une appréhension globale et synthétique de ce domaine. Ce phénomène est accentué par la diversité (et parfois l'inconsistance) des visions proposées par les différentes communautés de recherche. En effet, à l'exception du consensus constaté autour de l'infrastructure de base qui ne concerne que les couches basses de la pile des Web services (descriptions techniques pour assurer l'interopérabilité), des divergences de vues sur le rôle et le contenu des couches hautes de la pile (*e.g.*, les relations entre les Web services, les *business processes* et les workflows) apparaissent clairement dans la littérature. Ce point est important car il interpelle directement les problèmes d'intégration de processus d'entreprises, *i.e.* une intégration effectuée à un haut niveau d'abstraction en s'appuyant sur la sémantique des services. Ce type d'intégration constitue un des apports les plus prometteurs de l'approche

Web services. C'est la raison pour laquelle, dans la suite de cette section, nous présentons d'abord l'infrastructure de base des Web services. Nous abordons ensuite, à travers la notion de pile conceptuelle des Web services, les différents problèmes liés à la définition et la modélisation des contenus des couches hautes de cette pile.

Techniquement, un Web service peut donc être perçu comme étant une interface décrivant une collection d'opérations accessibles via le réseau à travers des messages XML standardisés. D'un point de vue technique, la description d'un Web service inclut tous les détails nécessaires à l'interaction avec le service comme, par exemples, le format des messages, les signatures des opérations, le protocole de transport et la localisation du service. Les Web services s'appuient sur des mécanismes et des protocoles standards et sont donc indépendants des langages de programmation (Java, J#, C++, Perl, C#, etc.), du modèle objet (COM, EJB, etc.) ainsi que des plates-formes d'implémentation (J2EE, .NET, etc.).

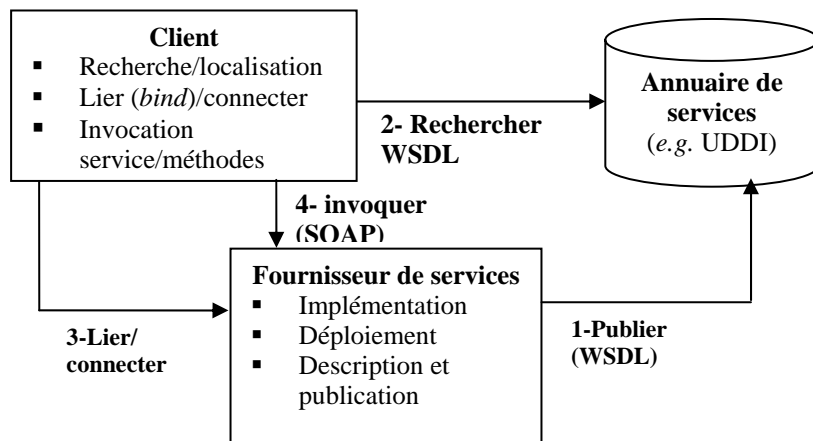
## 2.1 Architecture de référence

Les efforts de recherche et de développement récents autour des Web services ont conduit à un certain nombre de spécifications qui définissent aujourd'hui l'architecture de référence des Web services. Cette architecture vise trois objectifs importants <<http://www.w3.org/2002/ws/>> : (i) identification des composants fonctionnels, (ii) définition des relations entre ces composants et (iii) établissement d'un ensemble de contraintes sur chaque composant de manière à garantir les propriétés globales de l'architecture.

L'architecture de référence des Web services (cf. figure 1) s'articule autour des trois rôles suivants :

- Le fournisseur de service : correspond au propriétaire du service. D'un point de vue technique, il est constitué par la plate-forme d'accueil du service.
- Le client : correspond au demandeur de service. D'un point de vue technique, il est constitué par l'application qui va rechercher et invoquer un service. L'application cliente peut être elle-même un Web service.
- L'annuaire des services : correspond à un registre de descriptions de services offrant des facilités de publication de services à l'intention des fournisseurs ainsi que des facilités de recherche de services à l'intention des clients.

Les interactions de base entre ces trois rôles incluent les opérations de publication, de recherche et de liens (*bind*) d'opérations. Nous décrivons ci-dessous un scénario type d'utilisation de cette architecture. Le fournisseur de services définit la description de son service et la publie dans un annuaire de service. Le client utilise les facilités de recherche disponibles au niveau de l'annuaire pour retrouver et sélectionner un



**FIG. 1** – Architecture des web services.

service donné. Il examine ensuite la description du service sélectionné pour récupérer les informations nécessaires lui permettant de se connecter au fournisseur du service et d'interagir avec l'implémentation du service considéré.

Pour garantir l'interopérabilité des trois opérations précédentes (publication, recherche et lien), des propositions de standards ont été élaborées pour chaque type d'interactions. Nous citons, notamment les standards émergents suivants :

- SOAP définit un protocole de transmission de messages basé sur XML.
- WSDL introduit une grammaire commune pour la description des services.
- UDDI fournit l'infrastructure de base pour la publication et la découverte des services.

L'infrastructure de base autour de ces standards répond aux problèmes d'intégration technique des applications. En effet, contrairement aux approches d'intégration classiques qui ne sont pas exemptes d'inconvénients [50] (*e.g.*, les EAI qui sont des applications propriétaires), les Web services proposent une approche flexible et « universelle » pour l'intégration de systèmes hétérogènes en s'appuyant sur un modèle d'intégration basé sur un couplage faible des composants (*peer-to-peer*) et en exploitant de manière intensive les standards du Web. Ceci a pour effet de permettre une intégration des applications plus rapide et moins coûteuse et avec des perspectives d'évolution et de réutilisation réelles pour les entreprises.

Cependant, cette infrastructure n'est pas suffisante pour permettre une utilisation effective des Web services dans les domaines dont les exigences vont au-delà de la capacité d'interactions simples via des protocoles standards. Par exemple, dans le domaine du *e-business*, cette utilisation est motivée par les possibilités de coopération et de coordination entre des entreprises telles qu'on peut les percevoir dans la mise en œuvre de la gestion d'une chaîne logistique (eSCM) ou celle de la gestion des relations clients (eCRM). Le challenge est alors d'être capable de spécifier et de mettre en œuvre des business processes intra ou inter-entreprises. Ceci pose donc fondamentalement un problème d'intégration fonctionnelle des activités d'entreprises qui dépasse la simple capacité d'interactions via des protocoles standard. Pour des raisons de cohérence du discours, nous introduisons dans la section suivante la problématique de l'intégration inter-organisationnelle ainsi que ses concepts sous-jacents proposés dans la littérature.

## 2.2 Problématique de l'intégration

Les entreprises actuelles sont organisées en silos fonctionnels (ou par fonction) et s'avèrent, de fait, très inefficaces lorsqu'il s'agit de coordonner les flots de processus entre leurs différentes fonctions. La nécessité de les rendre réactives à leur environnement et performantes, les conduit de plus en plus à migrer vers une organisation transfonctionnelle (ou horizontale ou orientée processus) [17]. Le décroisement de ces silos revient en conséquence à définir des objectifs communs aux activités d'une entreprise : ce but peut être efficacement atteint via la spécification des processus opérationnels (business processes) de l'entreprise [52, 53], voire, dans le cas d'une entreprise étendue, des

processus opérationnels inter-entreprises ou inter-organisationnels<sup>1</sup> [29, 21]. Selon le consortium « Workflow Management Coalition » (WfMC – <<http://www.wfmc.org>>), un processus opérationnel est un ensemble de une ou plusieurs activités liées qui réalisent collectivement un objectif d'entreprise, normalement dans le contexte d'une structure organisationnelle définissant les rôles fonctionnels et les associations. Plus concrètement, au niveau organisationnel, un processus opérationnel est une structure logique indépendante des frontières établies par la hiérarchie de l'entreprise, qui exprime un regroupement d'activités ou de sous-processus réalisés par les différents départements (ou services) de l'entreprise. En d'autres termes, plutôt que de décrire les fonctions des services de l'entreprise en termes d'activités non nécessairement liées, on identifie un ensemble d'activités, issues de fonctions diverses, mais résumées autour de la réalisation d'un objectif commun. Via cette approche par processus, on a une vue dynamique de comment une organisation crée (ou non) de la valeur ajoutée. Mais la complexité des processus d'une entreprise est telle qu'il s'avère indispensable de les modéliser. Cette tâche peut elle-même s'avérer très complexe. C'est pourquoi la communauté des chercheurs d'Ingénierie d'Entreprise s'est efforcée, au cours des années 1990, de proposer des méthodologies de modélisation de processus d'entreprise intégrée implantées dans des architectures logicielles dont les plus reconnues sont ARIS [47, 48, 49], CIMOSA [53], PERA [54] ou encore GERAM [55]. Une fois les processus opérationnels d'une entreprise modélisés, l'étape suivante consiste à transcrire ces modèles en une forme (informatique) exécutable qui permet de suivre et de contrôler l'exécution des processus. Cette problématique de coordonner des activités issues de différentes fonctions autour d'un objectif commun, interpelle directement la notion de travail de groupe<sup>2</sup> supporté par un système de gestion de workflow (WfMS) ou *système workflow* ou encore système de gestion de processus (opérationnels). Un WfMS permet la définition et la création et la gestion de l'exécution des workflows qui automatisent (tout ou en partie) les processus opérationnels, ces derniers étant, dans les cas les plus complexes, des processus opérationnels inter-entreprises. La gestion d'un

---

<sup>1</sup> Un processus opérationnel inter-organisationnel est un processus organisationnel dont au moins une activité est sous-traitée par une organisation différente de celle qui est propriétaire de ce processus opérationnel.

<sup>2</sup> La notion de travail de groupe supporté par un outil de type *système workflow* est différente de celle de travail en groupe supporté par un outil de type collectif. La différence entre système workflow et collectif, est que les systèmes workflow prennent efficacement en charge la communication asynchrone entre les activités — c'est la prise en charge du processus en cours —, contrairement aux collectifs qui eux, privilégient l'aspect communication synchrone — c'est la prise en charge de l'activité en cours.



processus prend également en charge la coordination et la synchronisation des différents acteurs – ou agents : humains, machines autonomes ou applications logicielles – du processus (attribution à chaque acteur et au bon moment, des tâches dont il a la responsabilité avec les ressources — moyens inertes : machines, outils, véhicules... – et les informations qui lui sont nécessaires).

Il est donc fondamental, dans le cas de workflows intégrés, de bien situer leur place par rapport aux processus opérationnels. La technologie workflow s'appuie beaucoup, en effet, sur la technique de modélisation puisqu'un workflow se doit d'implanter un modèle en répondant, en particulier, aux questions suivantes :

- Quelles sont les activités à réaliser ?
- Quelles sont les compétences nécessaires pour réaliser ces activités ?
- Quand faut-il réaliser ces activités ?
- Quels sont les outils et les informations nécessaires à la réalisation de ces activités ?

En d'autres termes, pour espérer implanter efficacement un processus opérationnel dans un souci d'intégration via un workflow, il s'avère indispensable de modéliser ce processus avant de modéliser le workflow [5]. Les modèles de processus opérationnels relèvent, en effet, de l'intégration fonctionnelle, tandis que les modèles de workflows relèvent de l'intégration technique. Grosso modo, un système de gestion de workflow est aux processus opérationnels ce qu'un système de gestion de base de données est aux données ([51], p. 283). Le consortium WfMC propose la définition suivante d'un workflow : « un workflow est un processus d'une organisation, gérable par un outil workflow. Il est établi dans le but principal d'automatiser l'exécution du processus, mais il peut aussi servir à le simuler et à l'analyser ». Toujours selon le WfMC : « un système workflow définit, gère et réalise des procédures en exécutant des programmes dont l'ordre d'exécution est prédéfini dans une représentation informatique de la logique de ces procédures — les workflows ». Stohr & Zhao font le point sur les problèmes posés lors de l'automatisation des workflows. McCready [33] identifie trois catégories de systèmes workflows, à savoir :

- Les systèmes workflows administratifs (*General Purpose Workflow Management Systems*) qui implantent des processus administratifs. Les workflows obtenus sont répétitifs, à forte prédictibilité, à structure simple et sans grande complexité. De plus, ils subissent très rarement des modifications.

- Les systèmes workflows ad hoc qui permettent la gestion de processus non structurés ou peu structurés dans le sens où les tâches peuvent ne pas être connues lors de l'activation d'un workflow. Un processus non structuré est, par essence, très difficile à automatiser. Ils sont essentiellement utilisés pour les besoins d'un travail collaboratif, pour la co-décision, les acteurs étant les humains.
- Les workflows transactionnels permettent la gestion de processus et de tâches structurellement complexes (en particulier, les activités des processus sont réparties sur plusieurs sites). Ces processus sont à forte valeur ajoutée et réalisent le traitement d'un important volume de transactions. La mise en œuvre d'un workflow transactionnel nécessite l'interrogation de plusieurs systèmes d'information hétérogènes et distribués. De par leur orientation métier, les processus réalisés par ces workflows transactionnels, sont appelés à être fréquemment modifiés.

Dans la suite, nous ne nous intéressons qu'aux systèmes workflows transactionnels inter-organisationnels<sup>3</sup> [29] qui interpellent directement le contexte des Web services.

Stal [50] souligne qu'actuellement, les solutions pour résoudre les problèmes d'intégration technique d'entreprises s'appuient beaucoup sur la technologie EAI. Or, les solutions EAI sont, par essence, des solutions propriétaires, c'est-à-dire dédiées à la résolution de problèmes spécifiques, complexes à utiliser et qui ne peuvent pas bien interopérer les unes avec les autres. Par exemple, quand plusieurs entreprises intègrent des systèmes qui sont eux-mêmes intégrés en utilisant des EAI, les développeurs sont confrontés au problème récursif d'intégrer des solutions elles-mêmes intégrées. Dans un environnement très versatile où les intégrations fonctionnelle et technique doivent quasiment être réalisées au fil de l'eau, il est évident que la technologie EAI ne peut prétendre avoir l'ambition de s'imposer, ne serait-ce que parce qu'elle exige une forte composante humaine avec des temps de réaction très longs. Contrairement aux Web services qui intrinsèquement peuvent être conçus pour être indépendants des technologies hétérogènes des partenaires d'une organisation virtuelle.

On comprend alors mieux pourquoi l'infrastructure de base des Web services n'est pas suffisante pour répondre de manière satisfaisante à cette problématique de l'intégration. Cette dernière, en effet, exige, par essence, la définition d'un protocole qui permet aux activités intra et/ou inter entreprises composant un processus, d'être cohérentes relativement

---

<sup>3</sup> Un *workflow inter-organisationnel* est l'implantation d'un processus opérationnel inter-organisationnel.

à une organisation afin d'atteindre l'objectif visé. Il s'avère donc nécessaire d'étendre l'architecture de base des Web services comme présenté dans la section suivante.

### 2.3 Architecture étendue

Différentes extensions de l'architecture de référence ont été proposées dans la littérature. Le groupe architecture du W3C travaille activement à l'élaboration d'une architecture étendue standard.

Une architecture étendue est constituée de plusieurs couches se superposant les unes sur les autres, d'où le nom de pile des Web services. La figure 2 décrit un exemple d'une telle pile. La pile est constituée de plusieurs couches, chaque couche s'appuyant sur un standard particulier. On retrouve, au-dessus de la couche de transport, les trois couches formant l'infrastructure de base décrite précédemment. Ces couches s'appuient sur les standards émergents SOAP, WSDL et UDDI.

Comme mentionné précédemment, l'infrastructure de base définit les fondements techniques permettant de rendre les business processes accessibles à l'intérieur d'une entreprise et au-delà même des frontières d'une entreprise. Dans ce contexte deux types de couches permettent de la compléter : (i) les couches dites transversales [20] (*e.g.*, sécurité, administration, transactions et qualité de services (QoS)) rendent viable l'utilisation effective des Web services dans le monde industriel ; (ii) une couche Business processus permet l'utilisation effective des Web services dans le domaine du e-business. Dans la suite, nous nous intéresserons qu'à la couche business processus pour laquelle, on peut relever dans la littérature, les problèmes sous-jacents suivants :

- Comment les business processes peuvent-ils être représentés comme des Web services ?
- Nécessité de décrire comment les Web services sont utilisés pour implanter les activités d'un business process.
- Les problèmes de composition de service, *i.e.* quel(s) partenaire(s) va (vont) exécuter quelle(s) partie(s) d'un business process ?

Différents auteurs de la communauté de recherche s'accordent sur la nécessité de spécifier le comportement externe de chaque partie impliquée dans le protocole d'intégration de processus (partie publique) sans pour autant révéler leurs implémentations internes (partie privée). Deux raisons justifient cette séparation :

- Les entreprises ne tiennent pas forcément à révéler leurs prises de décisions internes et souhaitent préserver la confidentialité de leurs données.
- La séparation publique-privé permet de modifier la partie privée indépendamment de la partie publique.

A cet effet, différents langages ont été proposés dans le but de décrire le processus public d'un service (*e.g.*, WSCL) ou la spécification, de manière procédurale, de la composition de services (*e.g.*, BPML (<http://www.bpmi.org/>), BPEL4WS (<http://www106.ibm.com/developerworks/library/ws-bpel/>)).

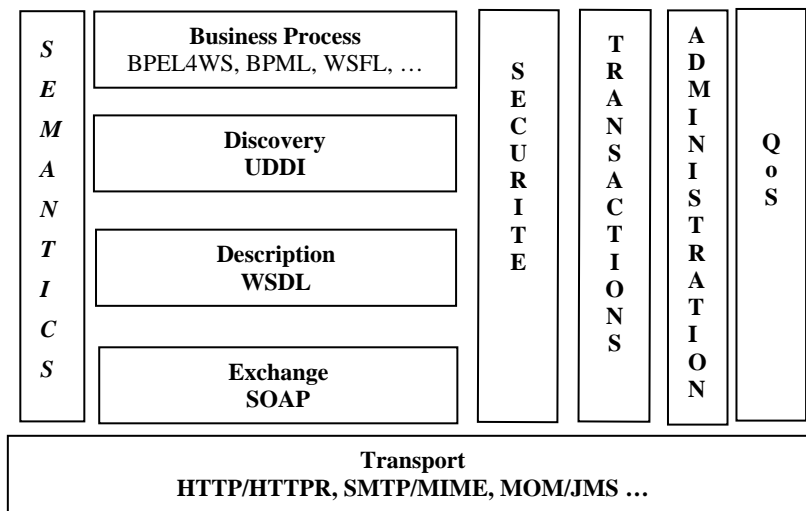


FIG. 3 – Pile des web services sémantiques

### 3 TRAVAUX ET RESULTATS EXISTANTS AUTOUR DES WEB SERVICES SEMANTIQUES ET RECHERCHES FUTURES

L'utilisation intensive des standards, d'une part, et celle du langage XML, d'autre part, constituent les deux caractéristiques fondamentales de la technologie des Web services. Par conséquent, les différentes spécifications d'un Web service peuvent être facilement traitées par une

machine. Par exemple, une description WSDL peut être automatiquement analysée par un parseur XML pour générer un *proxy* client (*i.e.* le code du *stub* client) facilitant ainsi la tâche du programmeur lors de l'implantation de son application. Cependant, le consensus sur la mécanique des interactions (*e.g.*, format des messages, types des données et protocoles d'échanges) n'est pas suffisant pour permettre aux Web services d'interagir de manière claire et non ambiguë. Par exemple, deux descriptions XML identiques peuvent avoir des significations différentes selon le contexte [44] <<http://www.w3.org/TR/webont-req>>. L'absence d'une sémantique explicite limite les possibilités d'automatisation des Web services [13, 15, 2]. Pour pallier cette limitation, il y a clairement un besoin de langages de descriptions des services qui permettent de conférer une signification explicite et non ambiguë aux descriptions des Web services. Par exemple, la nouvelle version du langage WSDL préconisé par le W3C permet de définir la sémantique des termes utilisés dans une description WSDL en référençant des ontologies RDF. La figure 3 reprend la pile des Web services augmentée d'une couche représentant la sémantique. Cette couche peut être exploitée par les trois couches horizontales supérieures (description, découverte et business processus) ainsi que par certaines couches verticales telles que les couches transactions ou QoS.

Notons qu'une condition *sine qua non* pour permettre une automatisation effective des Web services est que la sémantique de ces derniers soit décrite dans un langage qui la rend intelligible pour une machine.

De manière générale, la problématique de l'automatisation sous-tend deux classes de problèmes :

- Problèmes de modélisation : il s'agit d'abord d'identifier et de classer les connaissances pertinentes en fonction d'un ensemble de fonctionnalités visées (*e.g.*, découverte, composition, évaluation des performances, etc.) puis d'étudier ensuite comment les décrire de manière formelle et standard. Dans ce contexte, la notion d'ontologie en tant que conceptualisation formelle et consensuelle d'un domaine donné, peut jouer un rôle important pour associer une sémantique formelle à la description d'un Web service.
- Problèmes d'automatisation : il s'agit de développer des algorithmes et des mécanismes de raisonnement exploitant la sémantique des services pour automatiser les différentes fonctionnalités (*e.g.*, découverte de service, composition, évaluation de performances).

Nous décrivons ci-dessous quelques travaux de recherche relatifs à ces deux classes de problèmes.

## 3.1 Modélisation des services

### 3.1.1 DAML-S

DAML-S [13], [23] est une ontologie de Web services basée sur le langage DAML+OIL [26, 27]. DAML-S définit un ensemble de classes et de propriétés spécifiques à la description des services. Dans une description DAML-S, la *section profile* spécifie les informations relatives aux capacités d'un service. Elle consiste dans trois types d'informations :

- Une description du service compréhensible par les humains ;
- Le comportement fonctionnel du service représenté comme une transformation des paramètres d'entrée du service vers ses paramètres de sortie ;
- Plusieurs attributs fonctionnels qui spécifient des informations supplémentaires relatives au service (*e.g.*, prix du service).

Dans l'approche DAML-S, la section profile est utilisée à la fois par les fournisseurs pour publier leurs services et par les clients pour spécifier leurs besoins. Par conséquent, elle constitue l'information utile pour la découverte et la composition de services.

### 3.1.2 WSFM

WSFM est un cadre complet de description de services proposé dans le cadre du projet européen SWWS<sup>4</sup>. Il s'appuie sur les deux principes suivants :

- Découplage fort des divers composants qui réalisent une application de commerce électronique. Ceci implique une distinction claire entre les descriptions internes d'un service et ses descriptions publiques.
- Des mécanismes de médiation permettant un dialogue automatique entre les différents composants. De tels mécanismes incluent la médiation entre différentes terminologies ainsi que la médiation entre différents modèles d'interaction.

WSFM comprend quatre éléments principaux :

---

<sup>4</sup> <http://swws.semanticweb.org>

- des ontologies qui fournissent la terminologie utilisée par les autres éléments ;
- un répertoire d'objectifs qui définit les problèmes qui doivent être résolus par les Web services ;
- Des descriptions des Web services qui définissent les différents aspects liés aux Web services ;
- des médiateurs qui sont en charge des problèmes d'interopérabilité.

Dans le cadre du projet SWWS, WSFM devrait être complété par la définition d'un cadre pour la découverte de services ainsi que des mécanismes de médiation pour les Web services.

## 3.2 Problèmes d'automatisation

### 3.2.1 Découverte dynamique des services

On entend par découverte dynamique la possibilité de localiser automatiquement un Web service qui répond à des besoins particuliers. Différentes approches ont été proposées dans la littérature pour réaliser la découverte dynamique de services [3, 10, 19, 44, 2]. Toutes ces approches implantent en fait une découverte approximative car il n'est pas réaliste d'imaginer qu'il y a toujours un service qui correspond exactement aux besoins spécifiés. Ces approches diffèrent par le langage de description de services utilisé (*e.g.*, DAML-S, logique de description [1]...) et/ou par l'algorithme de découverte utilisé (matchmaking [44], test de subsumption [19], réécriture [2]). Par exemple, [3] propose d'utiliser des ontologies de processus pour décrire le comportement des services et définit un langage d'interrogation de processus (*Process Query Language*) pour interroger ces ontologies. [10] définit une ontologie basée sur le langage DAML pour décrire des ressources mobiles et proposent un processus de correspondance qui localise les ressources en fonction de leurs caractéristiques. Le processus de correspondance s'appuie sur des règles qui exploitent l'ontologie, les profils des services et la requête du client pour réaliser une correspondance à partir des relations attributs-valeurs. Un moteur de raisonnement Prolog permet la mise en œuvre d'un tel processus de correspondance. D'autres approches basées sur une description DAML-OIL des services proposent d'exploiter les mécanismes de raisonnement fournis par DAML-OIL pour supporter la découverte dynamique des Web services. González-Castillo *et al.* [19] décrivent une telle mise en œuvre dans laquelle l'algorithme de correspondance repose sur les tests

de subsumption et de consistance des descriptions. Paolucci *et al.* [44] proposent un algorithme de correspondance plus élaboré entre des services et des requêtes décrits en DAML-S. L'algorithme reconnaît différents degrés de correspondance qui sont déterminés par la distance minimale entre les concepts dans la taxonomie de concepts. De la même façon, le système ATLAS [45] opère sur des ontologies DAML-S et utilise deux ensembles séparés de filtres : (i) les attributs fonctionnels et (ii) les E/S des services. Enfin, le projet MKBEEM <<http://www.mkbeem.com>> exploite les logiques de description pour décrire les ontologies de services et définit un algorithme de découverte de service basé sur la notion de réécriture de concepts [22, 2].

### 3.2.2 Composition des services

L'objectif de la composition de service est de créer de nouvelles fonctionnalités en combinant des fonctionnalités offertes par d'autres services existants, composés ou non en vue d'apporter une valeur ajoutée. Étant donnée une spécification de haut niveau des objectifs d'une tâche particulière, la composition de service implique la capacité de sélectionner, de composer et de faire interopérer des services existants. Contrairement aux business processes « traditionnels » qui sont exécutés de manière prévisible et répétitive dans un environnement statique, les Web services composés s'exécutent dans un environnement versatile où le nombre de services disponibles évolue très rapidement. De plus, la forte compétition engendrée par la multitude de fournisseurs de services oblige les entreprises à adapter leurs services pour mieux répondre aux besoins des clients et ce à moindre coût. Comme le soulignent [9], ces deux facteurs imposent des contraintes fortes sur les systèmes qui délivrent des services composés. En conséquence, les business processes qui décrivent des services composés devront intégrer d'emblée ces contraintes en exhibant des possibilités réelles d'adaptabilité à leur environnement.

Des résultats concernant cette problématique commencent seulement à émerger. Les travaux existants s'intéressent à une modélisation abstraite des services et à la définition d'un cadre formel pour les composés [28], [4]. Des travaux récents de la communauté Web sémantique [42, 24] commencent à explorer des approches combinant des outils d'annotation de services et de planification de manière à pouvoir composer automatiquement des services en vue d'atteindre des fonctionnalités prédéfinies. Ce type d'approche constitue une alternative aux langages procéduraux de type BPEL4WS en permettant de générer l'implantation d'un service composite à partir de spécifications déclaratives de son



comportement. D'un autre côté, [28, 4] s'intéressent à la définition d'un cadre formel permettant de mieux comprendre les relations entre les propriétés globales d'un service composite et les propriétés locales de ses composants. La motivation étant de développer des techniques de vérification et de synthèse (construction) des propriétés d'un service composite à partir des propriétés des de ses composants. Tous ces travaux tentent de ré-exploiter et d'étendre des techniques existantes, telles que les logiques temporelles, l'algèbre des processus, les réseaux de Petri, la logique des situations, etc.

### 3.3 Évaluation des performances des Web services sémantiques

De manière générale, il existe deux types de critères de performance : ceux orientés utilisateur<sup>5</sup> et ceux orientés système<sup>6</sup> [35]. La QoS d'un workflow inter-organisationnel — comportant donc des services externes ou Web services — est un critère de performance orienté utilisateur. Puisque la QoS d'un tel workflow dépendra étroitement de celles des Web services qui le composent, on comprend aisément que la notion de qualité de Web services devient prédominante. Il convient de distinguer la QoS opérationnelle<sup>7</sup>, proposée *a priori* aux clients et mesurée sur le système réel, de la QoS qualitative et quantitative<sup>8</sup> elle aussi mesurée et exploitée *a posteriori* pour comprendre comment améliorer les performances d'un Web service offert. La problématique abordée ici concerne l'analyse *a priori* aussi bien qualitative<sup>9</sup> que quantitative<sup>10</sup> des performances d'un Web service. Il est clair que tout industriel raisonnable qui entrevoit l'apport potentiel des technologies de l'Internet

---

<sup>5</sup> Par exemple, le temps de réponse du système, c'est-à-dire le temps qui s'écoule entre la date de soumission d'un travail au système et celle de la fin d'exécution de cette tâche

<sup>6</sup> Par exemple, le débit du système qui mesure le taux de productivité du système relativement à sa charge, ou encore les taux d'occupation de ses ressources qui aident à identifier les goulets d'étranglement du système

<sup>7</sup> Sécurité, intégrité des informations, traçabilité des opérations, audits de sécurité, contrôle, filtrage et supervision des accès

<sup>8</sup> Délai, gigue ou variation du délai de bout en bout, débit maximum atteint, disponibilité ou taux moyen d'erreurs d'une liaison

<sup>9</sup> Absence de verrous mortels dans un web-service, validation d'un web-service ou test que le web-service se comporte comme attendu, vérification de l'existence de certaines propriétés d'un web-service

<sup>10</sup> Durée moyenne de réalisation d'un web-service, taux moyen d'échecs d'un web-service, coût moyen d'un web-service

en termes de bénéfices, n'acceptera d'y souscrire que s'il est capable, entre autres, d'évaluer *a priori* et objectivement les performances qualitatives et/ou quantitatives des Web services (complexes ou non) qu'il souhaite proposer.

Concernant l'évaluation qualitative d'un workflow, ce sont les chercheurs de la communauté des réseaux de Petri [14] qui sont le plus actifs. L'un des résultats des plus intéressants est donné dans [42] puisqu'il propose des solutions pour décrire, simuler, composer automatiquement, tester et vérifier la composition de Web services décrits avec l'ontologie DAML-S. Ils s'intéressent donc — et c'est ce qui est novateur — à des Web services sémantiques. Un processus composite décrit avec DAML-S, est traduit dans un langage de la logique du premier ordre, puis cette spécification est elle-même représentée sans ambiguïté par un réseau de Petri dont les propriétés sont étudiées par simulation. Cependant, même si les auteurs évoquent le problème de l'évaluation des performances de la QoS d'un Web service, il faut bien avouer que cet objectif leur est actuellement inaccessible, ne serait-ce que parce que les réseaux de Petri considérés n'intègrent ni le temps, ni le partage des ressources informatiques exécutant les Web services, la prise en compte de ces ressources étant prévue comme une extension de DAML-S.

Comme le souligne à juste titre [8], si la QoS a été une préoccupation majeure, sinon fondamentale, de domaines tels que les réseaux [12, 18], les applications temps-réel [11] et le middleware [16, 25], peu de groupes de chercheurs se sont efforcés d'intégrer le concept de QoS dans les systèmes de gestion de workflows à l'exception, peut-être, de [38, 43, 56]. En conséquence, très peu de travaux de recherche traitent du problème de l'évaluation des performances quantitatives de la QoS d'un workflow et, a fortiori, d'un workflow inter-organisationnel (donc de Web services). Or, le concept de QoS d'un workflow est devenu aujourd'hui incontournable pour certifier un workflow en termes de performances qualitative et quantitative, comme le montrent la création de modèles de qualité proposés par la norme ISO 8402. Selon les auteurs, la QoS d'un workflow représente les caractéristiques qualitatives et quantitatives d'une application workflow nécessaires pour atteindre un ensemble d'objectifs prédéfinis. Toujours selon les auteurs, cette situation est due, à l'absence de mécanismes dans les WfMS actuels permettant de spécifier la QoS d'un workflow. Ils proposent donc un modèle de QoS d'un workflow inter-organisationnel construit en définissant des métriques QoS des tâches pouvant composer un tel workflow. Les métriques QoS d'une tâche jugées pertinentes sont le temps et le coût d'une tâche, ainsi que la fiabilité et la fidélité d'une tâche.

Force est de constater que l'exploitation d'un modèle d'évaluation des performances d'un système, nécessite de disposer *a priori* des caractéristiques des composants du modèle puisque ces caractéristiques sont les données d'entrée du modèle. En ce qui concerne les Web services, il faut connaître (au moins) des durées (moyennes) des activités, ainsi que la manière dont ces activités s'enchaînent, cet enchaînement pouvant être spécifié par des probabilités (dont les valeurs sont généralement supposées être indépendantes du temps). L'obtention des caractéristiques d'un Web service n'est pas triviale puisque le fait qu'un Web service soit exécuté par un fournisseur, conduit le client à une quasi-impossibilité de connaître *a priori* ces caractéristiques. Une solution consiste alors à effectuer des mesures sur le système pour tenter, à l'aide de techniques statistiques, telles que celles proposées par [39, 40], d'estimer ces caractéristiques, puis de les injecter dans un modèle d'évaluation. Cardoso *et al.* [8] ont exploité cette technique pour alimenter le modèle QoS qu'ils proposent. Brièvement, leur modèle fournit une approche multidimensionnelle pour la découverte de Web services et leur intégration à l'aide de métriques syntaxiques, sémantiques et comporte des métriques opérationnelles principales (ou dimensions) qui sont le temps, le coût, la fiabilité et la fidélité, ces métriques étant calculées automatiquement. Selon les auteurs, ce modèle QoS offre la possibilité de construire des e-workflows interopérables grâce, en particulier, à l'utilisation d'ontologies pour décrire les interfaces entre les tâches d'un e-workflow et les Web services exploités par ce e-workflow. Le modèle QoS proposé permet également de calculer la QoS d'un e-workflow à partir des attributs (ou métriques) de ses composants selon deux techniques : (i) en utilisant une modélisation mathématique [6, 7], qui est une méthode de réduction de workflows stochastiques qui consiste à appliquer un ensemble de règles de réduction à un workflow jusqu'à ce que ce workflow se réduise à une tâche atomique [31], (ii) en utilisant la simulation aléatoire à événements discrets [41, 36, 37]. Le choix d'une de ces techniques dépend essentiellement du compromis entre le temps de calcul des QoS d'un e-workflow et la qualité des résultats obtenus.

Dans [30, 29], les auteurs proposent d'évaluer les performances quantitatives d'un Web service avec des chaînes de Markov à temps continu et espace d'état discret. Le problème (fondamental) pour modéliser un service inter-organisationnel et en accord avec [8], est que l'on ne le maîtrise pas explicitement. Il est donc nécessaire de collecter des informations concernant ce service pendant son exécution (ce service étant réalisé par un fournisseur de services). Une hypothèse forte est alors que le comportement observé d'un service représente son comportement futur (l'objectif étant de faire des prédictions). Dans ces travaux, un

service est constitué d'un ensemble de tâches qui pourront être activées lors d'une exécution de ce service. Les auteurs définissent l'état d'un service (observé) comme l'ensemble des tâches actives d'un service à un instant donné. L'exécution d'un service est alors modélisée par une chaîne de Markov dont les états sont obtenus à partir du journal de ce service et dont les caractéristiques sont calculées par un processus d'agrégation des données [29].

Dans le contexte des Web services sémantiques, la motivation de recherche réside dans la possibilité d'intégrer, dans un modèle d'évaluation des performances d'un Web service, des aspects sémantiques permettant d'identifier de nouveaux critères de qualité d'un Web service et de retenir ces critères dans la mise en œuvre effective d'un Web service pour la découverte et la composition dynamique de Web services.

L'aspect sémantique pour les Web services n'est actuellement considéré par les chercheurs qu'au niveau des tâches d'un e-workflow. Il serait peut-être opportun d'étendre ce concept de sémantique à d'autres fonctions de l'entreprise, c'est-à-dire de réfléchir à quels peuvent être les impacts, par exemple financiers, des Web services sémantiques. Cela reviendrait alors à raisonner, non plus sur des critères de performances d'un e-workflow, mais sur des indicateurs de performances perçus comme pertinents par les décideurs d'une entreprise, indicateurs bien sûr qu'il reste aujourd'hui à identifier et à construire. Pour y parvenir, nous pensons qu'il devra être nécessaire d'intégrer, entre autres, la notion d'échelle de temps dans les ontologies des métriques QoS des Web services sémantiques et d'étudier leurs conséquences sur les problèmes d'évaluation qui ne manqueront pas de surgir lorsque l'échelle de temps des indicateurs et celles des critères se rapprocheront.

## **4 CONCLUSION**

Aujourd'hui, les Web services sémantiques constituent une voie prometteuse permettant de mieux exploiter les Web services en automatisant, autant que possible, les différentes tâches liées au cycle de vie d'un service. Ils apparaissent donc indispensables pour permettre une utilisation effective des Web services dans des applications industrielles (*e.g.*, problème du passage à l'échelle, versatilité du Web, etc.). Ils posent aujourd'hui un certain nombre de problèmes, qui interpellent différentes communautés de recherche, aussi bien théoriques qu'appliqués. Le nombre de nouvelles revues, le volume important de publications et de

projets dédiés à ce thème dénotent une vitalité réelle de ce domaine de recherche émergent.

Cependant, on remarque que la tendance actuelle des communautés de recherche s'intéressant aux Web services sémantiques est de ne pas tenir compte explicitement des caractéristiques fondamentales des Web services et de l'environnement dans lequel ils doivent s'intégrer (*e.g.*, les e-workflows). A notre avis, le succès de cette voie de recherche dépendra étroitement de sa capacité, entre autres, à tenir compte des facteurs suivants :

- Les travaux de recherche devront intégrer le plus possible les caractéristiques des futurs standards actuellement en cours d'élaboration, les éditeurs de logiciels (*e.g.*, IBM, Microsoft...) étant fortement impliqués dans cette tâche. Ils doivent donc s'efforcer d'exploiter/compléter ces futurs standards et non pas ignorer leur existence ou les concurrencer. De la même manière, il est important de bien identifier les contraintes imposées par les fonctions d'entreprise afin de resituer les problématiques de recherche.
- La volonté d'automatiser à outrance n'est certainement pas une voie réaliste. Certains travaux de recherche semblent faire abstraction de la complexité du contexte de l'intégration de par les hypothèses simplificatrices fortes qu'ils imposent dans leurs solutions. En effet, le contexte de l'intégration fonctionnelle est tel que de nombreuses tâches doivent rester à la charge d'humains. Il est, par exemple, illusoire de vouloir automatiser complètement la gestion d'une chaîne logistique. Par ailleurs, comme le souligne [46] pour les workflows, la dimension temps est une composante fondamentale à prendre en compte pour l'automatisation des Web services sémantiques. Dans ce sens, l'intégration d'un Web service sémantique dans un e-workflow n'a de sens que s'il est pertinent et réaliste d'automatiser les tâches de ce Web service de sorte à apporter de la valeur ajoutée au e-workflow.
- Le concept de sémantique tel que défini dans le contexte du Web sémantique – *i.e.*, décrire la sémantique de manière à la rendre intelligible pour les machines –, semble trop restrictif. En effet, il est également très important d'explicitier la sémantique des Web services en vue de faciliter leur utilisation par les humains, même pour les situations où l'automatisation ne semble pas réaliste. Il est notoire que dans le domaine des bases de données par exemple, les modèles sémantiques (*e.g.*, le modèle Entité/Association de Chen) ont été proposés à l'origine pour faciliter la compréhension de la sémantique des données d'un système d'information par les humains. Ces modèles

se sont avérés très utiles par la suite pour automatiser partiellement le processus de conception d'une base de données.

## 5 RÉFÉRENCES

- [1] F. Baader , D. Calvanese, D. McGuinness, D. Nardi & P.F. Patel-Schneider, (editors) (2003). *The Description Logic Handbook. Theory, Implementation and Applications*, Cambridge University Press, 574 pages.
- [2] B. Benatallah , M-S. Hacid, C. Rey & F. Toumani (2003). Semantic Reasoning for Web Services Discovery, *WWW Workshop on E-Services and the Semantic Web*, Budapest, Hungary.
- [3] A. Bernstein & M. Klein (2002). Discovering Services: Towards High Precision Service Retrieval. In *CaiSE workshop on Web Services, e-Business, and the Semantic Web: Foundations, Models, Architecture, Engineering and Applications*. Toronto, Canada.
- [4] T. Bultan, X. Fu, R. Hull & J. Su (2003). Conversation specification: a new approach to design and analysis of e-service composition. *WWW 2003*: 403-410.
- [5] C. Bussler (1996). Workflow-Management-Systems as enterprise engineering tools, in *Modelling and methodologies for enterprise integration*, Edited by Bernus, P. and Nemes, L., 234-247, Chapman & Hall.
- [6] J. Cardoso (2002a). Stochastic Workflow Reduction Algorithm. *LSDIS Lab, Department of Computer Science*, University of Georgia.
- [7] J. Cardoso (2002b). Workflow Quality of Service and Semantic Workflow Composition. *Ph.D. Dissertation*. Department of Computer Science, University of Georgia, Athens, GA.
- [8] J. Cardoso, J. Miller, A. Sheth & J. Arnolf (2002). Modeling Quality of Service for workflows and Web Service Processes. *Technical Report# 02-2002, LSDIS Lab. Computer Science*, University of Georgia, 44 pages.
- [9] F. Casati & M-C. Shan (2001). Models and Languages for Describing and Discovering E-Services. In *ACM SIGMOD*, Santa Barbara, USA.
- [10] D. Chakraborty, F. Perich, S. Avancha, & A. Joshi (2001). DReggie: Semantic Service Discovery for M-Commerce Applications. In *Workshop on Reliable and Secure Applications in Mobile Environment, 20th Symposium on Reliable Distributed Systems*, pages 28–31.
- [11] D. Clark, S. Shenker & L. Zhang (1992). Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism. *Proceedings of ACM SIGCOMM*, pp. 14-26.

- [12] R.L. Cruz (1995). Quality of service guarantees in virtual circuit switched networks. *IEEE J. Select. Areas Commun.* 13(6): p.1048-1056.
- [13] DAML Services Coalition (2002). DAML-S: Web Service Description for the Semantic Web. In *The First International Semantic Web Conference (ISWC)*, pages 348–363.
- [14] M. Diaz (2001). *Les réseaux de Petri*. Hermès.
- [15] D. Fensel, C. Bussler, & A. Maedche (2002). Semantic Web Enabled Web Services. In *International Semantic Web Conference*, Sardinia, Italy, pages 1–2.
- [16] S. Forlund & J. Koistinen (1998). Quality-of-Service Specification in Distributed Object Systems. *Distributed Systems Engineering Journal* 5(4).
- [17] D.A. Garvin (1998). The Processes of Organization and Management. *Sloan Management Review*, Summer, 33-50.
- [18] L. Georgiadis, R. Guerin, V. Peris & K. Sivarajan (1996). Efficient QoS Provisioning Based on Per Node Traffic Shaping. *IEEE ACM Transactions on Networking* 4(4):482-501.
- [19] J. González-Castillo, D. Trastour, & C. Bartolini (2001). Description Logics for Matchmaking of Services. In *KI-2001 Workshop on Applications of Description Logics* Vienna, Austria, Sep. <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-44/>.
- [20] K. D. Gottschalk, S. Graham, H. Kreger & J. Snell (2002). Introduction to Web services architecture. *IBM Systems Journal* 41(2): 170-177.
- [21] P. Grefen, K. Aberer, Y. Hoffner & H. Ludwig (2000). CrossFlow: cross-organisational workflow management in dynamic virtual enterprises. *International Journal of Computer Systems Science & Engineering*, vol. 15, N°5, 277-290.
- [22] M.S. Hacid, A. Léger, C. Rey & F. Toumani (2002). Dynamic discovery of e-services: A description logics based approach, *BDA'2002*, Paris, 21–25.
- [23] J. Hendler & D. L. McGuinness (2000). The DARPA Agent Markup Language. *IEEE Intelligent Systems*, 15(6):67–73.
- [24] J. Hendler, D. Nau, B. Parsia, E. Sirin & D. Wu (2003). Automating DAML-S Web Services Composition Using SHOP2. in *ISWC'03*, p.p. 195-210.
- [25] M.A. Hiltunen, R. Schlichting, C.A. Ugarte & G.T. Wong (2000). Survivability through Customization and Adaptability: The Cactus Approach. *DARPA Information Survivability Conference and Exposition (DISCEX 2000)*, pp. 294-307.
- [26] I. Horrocks, P.F. Patel-Schneider, & F. van Harmelen (2002a). Reviewing the Design of DAML+OIL: An Ontology Language for the Semantic

- Web. In *Proc. of the 18th Nat. Conf. on Artificial Intelligence (AAAI)*, 792-797.
- [27] I. Horrocks (2002b). DAML+OIL: A Reasonable Web Ontology Language. In *Proc. of the EDBT'2002 Prague*, Czech Republic, pages 2–13.
- [28] R. Hull, M. Benedikt, V. Christophides & J. Su (2003). E-services: a look behind the curtain. *PODS 2003*, 1-14.
- [29] J. Klingemann, J. Wäsch & K. Aberer (1998). Adaptive outsourcing in cross-organizational workflows. *GMD Report 30*, GMD – German National Research Center for Information Technology.
- [30] J. Klingemann, J. Wäsch & K. Aberer (1999). Deriving Service Models in Cross-Organizational Workflows. *Procs. 9th Int. Worksh. Research Issues in Data Eng.*, Sydney, Australia.
- [31] K.J. Kochut, A.P. Sheth & J.A. Miller (1999). ORBwork: A CORBA-Based Fully Distributed, Scalable and Dynamic Workflow Enactment Service for METEOR. *Large Scale Distributed Information Systems Lab, Department of Computer Science*, University of Georgia, Athens, GA.
- [32] E. M. Maximilien & M.P. Singh (2001). Conceptual Model of Web Service Reputation, *SIGMOD Record 31(4)*: 36-41
- [33] McCready (1992). There is more than one kind of workflow software. *Computerworld, November 2*: 86-90.
- [34] S. McIlraith, T.C. Son, & H. Zeng (2001). Semantic Web Services. *IEEE Intelligent Systems. Special Issue on the Semantic Web*, 16(2):46–53.
- [35] D.A. Menascé & V.A.F. Almeida (1998). *Capacity Planning for WEB PERFORMANCE*. Prentice Hall.
- [36] J.A. Miller, R. Nair, Z. Zhang & H. Zhao (1997). JSIM: A Java-Based Simulation and Animation Environment. *Proceedings of the 30th Annual Simulation Symposium, Atlanta, GA.*, 786-793.
- [37] J.A. Miller, A.F. Seila & X. Xiang (2000). The JSIM Web-based Simulation Environment. *Future Generation Computer Systems: Special Issue on Web-Based Modeling and Simulation*, 17(2): 119-133.
- [38] A.V. Moorsel (2001). Metrics for the Internet Age: Quality of Experience and Quality of Business. *Technical Report HPL-2001-179, HP Labs, August 2001. Also published in 5th Performability Workshop, Erlangen, Germany.*
- [39] J.D. Musa (1993). Operational Profiles in Software-Reliability Engineering. *IEEE Software*, 10(2): 14-32.
- [40] J.D. Musa (1999). *Software reliability engineering: more reliable software, faster development and testing*. New York, McGraw-Hill.
- [41] R. Nair, J.A. Miller & Z. Zhang (1996). A Java-Based Query Driven Simulation Environment. *Proceedings of the 1996 Winter Simulation Conference, Colorado, CA.*, 786-793.



- [42] S. Narayanan & S. McIlraith (2002). Simulation, Verification and Automated Composition of Web Services, *Eleventh International World Wide Web Conference (WWW2002)*, Honolulu.
- [43] J. O'Sullivan, D. Edmond & A. ter Hofstede (2002). What's in a Service? *Distributed and Parallel Databases*, 1 2(2-3), p. 117-133.
- [44] M. Paolucci, T. Kawamura, T.R. Payne, & K.P. Sycara (2002). Semantic Matching of Web Services Capabilities. *In Int. Semantic Web Conference, Sardinia, Italy*, pages 333–347.
- [45] T.R. Payne, M. Paolucci, & K. Sycara (2001). Advertising and Matching DAML-S Service Descriptions (position paper). *In International Semantic Web Working Symposium, Stanford University, California, USA*.
- [46] H.A Reijers (2003). *Design and control of workflow processes*, LNCS 2617, Springer-verlag, Berlin Heidelberg, 1-29.
- [47] A.-W. Scheer (1992). *Architecture of Integrated Information Systems*. Springer-Verlag, Berlin.
- [48] A.-W. Scheer (1999). *ARIS – Business Process Frameworks*. Third Edition, Springer-Verlag, Berlin.
- [49] A.-W. Scheer (2000). *ARIS – Business Process Modeling*. Third Edition, Springer-Verlag, Berlin.
- [50] M. Stal (2002). Web Services: Beyond Component-Based Computing, *Com. of The ACM*, 45(10), 71-76.
- [51] E.A Stohr & J.L. Zhao (2001). Workflow Automation: Overview and Research Issues. *Information Systems Frontiers*, Kluwer Academic Publishers, 3:3, 281-296.
- [52] F.B. Vernadat (1993). CIMOSA: Enterprise Modelling and Integration Using a Process-based Approach, *In: DIISM*.
- [53] F.B. Vernadat (1996). *Enterprise modelling and integration: principles and applications*, Editions Chapman & Hall.
- [54] T.J. Williams (1994). The Purdue Enterprise Reference Architecture. *Computers in Industry*, 24(2-3), 141, 158.
- [55] T.J. Williams (1995). Development of GERAM, a Generic Enterprise Reference Architecture and Enterprise Integration Methodology. *In Integrated Manufacturing Systems Engineering* (P. Ladet and F.B. Vernadat, eds), Chapman & Hall, London, UK, 279-288.
- [56] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam & Q. Sheng (2003). Quality-driven Web Service Composition. *In Proc. of 14th International Conference on World Wide Web (WWW'03)*, Budapest, Hungary, p. 411-421, ACM Press.