



CIRANO
Centre interuniversitaire de recherche
en analyse des organisations

RAPPORT BOURGOGNE

**LES *WEB SERVICES* ET LEUR
IMPACT SUR LE COMMERCE B2B**

Gilbert Babin

Fellow, CIRANO

Michel Leblanc

M.Sc. commerce électronique
Associé, Adviso Conseil Inc.

(Août 2003)

2003RB-07

Les Rapports bourgogne / The Burgundy Reports

Documents de synthèse portant sur des questions d'intérêt général produits par des Fellows CIRANO, les Rapports bourgogne contribuent à alimenter la réflexion et le débat public sur des questions d'actualité. Les idées et les opinions émises dans ces rapports sont sous l'unique responsabilité des auteurs, et ne représentent pas nécessairement les positions du CIRANO ou de ses partenaires corporatifs, universitaires et gouvernementaux.

The Burgundy Reports are written by CIRANO Fellows on issues of general interest, and aim at encouraging discussion and debate. The observations and viewpoints expressed are the sole responsibility of the authors; they do not necessarily represent positions of CIRANO or its corporative, university or governmental partners.

CIRANO

Le CIRANO est un organisme sans but lucratif constitué en vertu de la Loi des compagnies du Québec. Le financement de son infrastructure et de ses activités de recherche provient des cotisations de ses organisations-membres, d'une subvention d'infrastructure du Ministère du développement économique et régional, de même que des subventions et mandats obtenus par ses équipes de recherche.

CIRANO is a private non-profit organization incorporated under the Québec Companies Act. Its infrastructure and research activities are funded through fees paid by member organizations, an infrastructure grant from the Ministère du développement économique et régional and grants and research mandates obtained by its research teams.

Les organisations-partenaires / The Partner Organizations

HEC Montréal
École Polytechnique
Université Concordia
Université de Montréal
Université du Québec à Montréal
Université Laval
Université McGill
Ministère des Finances du Québec
Ministère du développement économique et régional (MDER)
Alcan inc.
AXA Canada
Banque du Canada
Banque Laurentienne du Canada
Banque Nationale du Canada
Banque Royale du Canada
Bell Canada
BMO Groupe Financier
Bombardier
Bourse de Montréal
Développement des ressources humaines Canada [DRHC]
Fédération des caisses Desjardins du Québec
Gaz Métropolitain
Hydro-Québec
Industrie Canada
Pratt & Whitney Canada Inc.
Raymond Chabot Grant Thornton
Ville de Montréal

© 2003 Gilbert Babin et Michel Leblanc. Tous droits réservés. All rights reserved.
Reproduction partielle permise avec citation du document source, incluant la notice ©.
Short sections may be quoted without explicit permission, provided that full credit, including © notice, is given to the source.

1 Introduction

Le commerce électronique peut être défini comme étant l'utilisation des télécommunications et des systèmes d'information pour effectuer des transactions commerciales entre entreprises et individus. Avec l'essor d'Internet, il devient de plus en plus intéressant pour les entreprises d'utiliser un support électronique pour leurs transactions commerciales. Cependant, cette utilisation accrue d'Internet ne se fait pas sans heurts. Plusieurs problèmes restent à résoudre afin de faciliter l'utilisation de l'Internet et de profiter pleinement des bénéfices du commerce électronique.

Considérons un instant les difficultés associées aux échanges électroniques entre entreprises. Pour qu'un message reçu soit correctement interprété par les applications d'affaires, il faut dans un premier temps que la syntaxe du message soit reconnue par le logiciel recevant ce message. Cela peut sembler simple à prime abord, mais demande que les deux parties s'entendent sur la façon de représenter toutes les valeurs transmises dans le message. Dans un deuxième temps, les deux entreprises doivent s'entendre sur la signification de chaque champ du message. Une erreur d'interprétation peut s'avérer critique. Par exemple, une mesure en pouce interprétée en centimètres pourrait provoquer un bris d'équipement majeur. Finalement, le contexte d'affaire dans lequel s'insère ce message doit clairement être défini.

Le problème de l'échange électronique d'information entre entreprises n'est pas nouveau. Parmi les solutions développées pour résoudre ce problème, on retrouve la norme EDI (*Electronic Data Interchange*)¹. Datant des années 1980, cette norme requiert des entreprises d'adopter des structures de messages très strictes et peu flexibles. De plus, les logiciels nécessaires pour utiliser cette norme sont très coûteux. Finalement, toutes les transactions se font sur des réseaux privées ayant des coûts d'utilisation élevés. L'essor d'Internet a été vu par plusieurs comme une façon de réduire drastiquement les coûts liés à l'EDI. Cependant, une des difficultés majeures de l'EDI demeure la structure stricte des messages qui est particulièrement complexe pour les moyens techniques et financiers des PME.

Le standard XML (*eXtensible Markup Language*), descendant direct de SGML (*Standard Generalized Markup Language*) et parent de HTML (*Hypertext Markup Language*), est selon plusieurs une alternative intéressante à EDI². Il offre une syntaxe de messages standardisée tout en permettant de redéfinir la structure des messages au besoin. Ainsi, XML définit une famille de langages partageant une

« Avec l'essor d'Internet, il devient de plus en plus intéressant pour les entreprises d'utiliser un support électronique pour leurs transactions commerciales. Cependant, cette utilisation accrue d'Internet ne se fait pas sans heurts. »

¹ UN/EDIFACT Draft Directory. *United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport* (UN/EDIFACT).
<http://www.unece.org/trade/untdid/welcome.htm>.

² B. Meltzer and R. Glushko. *XML and electronic commerce: enabling the network economy*, ACM SIGMOD Record, 27(4):21-24, Dec. 1998.

syntaxe simple commune. À l'instar de SGML et HTML, la syntaxe de XML est facilement lisible par un utilisateur. Contrairement à HTML qui impose un ensemble de balises, XML permet à l'utilisateur d'utiliser des balises pré-définies ou de définir ses propres balises.

Par ailleurs, le commerce électronique ne se limite pas à l'échange de messages. Les entreprises permettent à leurs partenaires d'affaires d'accéder directement aux fonctions de leur système d'information. On parle alors d'intégration entre les différents systèmes d'information. Plusieurs standards permettent aux entreprises de réaliser cette intégration, notamment CORBA (*Common Object Request Broker Architecture*)³, DCOM (*Distributed Component Object Model*)⁴ et RMI (*Remote Method Invocation*)⁵. Cependant, ces solutions requièrent des efforts non négligeables afin de bien coupler les différents systèmes d'information. Ainsi, il est souvent difficile, voire impossible, aux applications développées à l'aide de ces approches de traverser les coupe-feu d'entreprises. De plus, aucune de ces solutions ne jouit d'une acceptation généralisée. des manufacturiers informatiques.

Une alternative à CORBA, DCOM et RMI, basée sur XML, a récemment vu le jour. Il s'agit des *Web Services*. Cette solution au problème d'intégration semble très prometteuse. Ce document se veut une introduction aux *Web Services*. Il s'adresse particulièrement aux gestionnaires ayant à faire des choix technologiques et aux gestionnaires des technologies de l'information, mais il sera sans doute aussi très utile aux technologues encore peu familiers avec la technologie des *Web services*. Nous montrerons donc ce qu'ils sont et comment ils peuvent résoudre bon nombre de problèmes liés au commerce électronique, plus particulièrement le B2B (*Business to Business*).

2 Les Web Services

Les *Web Services* sont des applications qui relient des programmes, des objets, des bases de données ou des processus d'affaires à l'aide de XML et de protocoles Internet standards. Les *Web Services* sont des compléments aux programmes et applications existantes, développées à l'aide de langages tel que Visual Basic, C, C++, C# (*C sharp*), Java ou autre, et servent de pont pour que ces programmes communiquent entre eux.

Les *Web Services* définissent non seulement les données transmises entre deux applications, mais aussi comment traiter ces données et les relier à l'interne et à

« Les Web Services sont des applications qui relient des programmes, des objets, des bases de données ou des processus d'affaires à l'aide de XML et de protocoles Internet standards. »

³ <http://www.corba.org/>

⁴ DCOM Technical Overview, <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/default.asp>.

⁵ Java Remote Method Invocation - Distributed Computing For Java, <http://java.sun.com/marketing/collateral/javarmi.html>.

l'externe d'une application logicielle sous-jacente⁶. De ce fait, les *Web Services* permettent aux entreprises et individus de publier des liens vers leurs applications de la même manière qu'ils publient des liens vers leurs pages Web. Conséquemment, les *Web Services* peuvent faire en sorte que toute les ressources informatiques dont une entreprise a besoin soient des ressources distribuées à la grandeur de l'Internet⁷.

Contrairement à la plupart des applications de type client/serveur⁸, les *Web Services* ne fournissent pas d'interface usager. Ils sont utilisés afin d'envoyer des données destinées à être lues par des machines⁹. Cependant, les programmeurs peuvent tout de même développer une interface graphique pour l'utilisateur, auxquels ils pourront ajouter une panoplie de *Web Services* afin de personnaliser une page Web ou pour offrir une fonctionnalité spécifique à des utilisateurs.

« ... les Web Services peuvent faire en sorte que toute les ressources informatiques dont une entreprise a besoin soient des ressources distribuées à la grandeur de l'Internet. »

2.1 Éléments de définition

Plusieurs acteurs définissent les *Web Services* par des caractéristiques technologiques distinctives. Certains éléments principaux nous apparaissent importants :

- Une application logicielle identifiée par un URI (*Uniform Resource Identifier*)

Un *Web Service* est une application autosuffisante en ce sens qu'elle effectue une seule tâche et que ses composantes décrivent ses propres entrées et sorties de telle sorte que d'autres logiciels qui invoquent le *Web Service* puissent interpréter ce qu'il fait, comment invoquer sa fonctionnalité et à quel résultat cet autre service peut s'attendre.

Les *Web Services* sont, en quelque sorte, le prolongement de la programmation objet. Ainsi, un *Web Service* est donc une sorte d'objet avec une seule fonctionnalité permettant, avec d'autres *Web Services*, la composition d'une application plus large pouvant avoir plusieurs fonctionnalités. En fait, c'est l'une des briques d'un mur qui en comporte plusieurs.

Un URI est la façon d'identifier un point de contenu sur le Web, que ce soit un fichier texte, audio ou vidéo. L'URI le plus connue est l'adresse d'une page Web, par exemple : <http://www.cirano.qc.ca>. Le *Web Service* est donc accessible en spécifiant son URI.

- Capacité des interfaces et liaisons (*bindings*) d'être publiées, localisées et invoquées via XML

⁶ Résumé et traduit librement de Eric Newcomer, *Understanding Web Services : XML, WSDL, SOAP and UDDI*, Addison-Wesley, 2002, pp. 3-15.

⁷ *The SMB Internet Scenario*, Gartner Research, 2001, COM-13-4497.

⁸ *Web Services* - Webopedia.com, http://www.webopedia.com/TERM/W/Web_services.html

⁹ New Public Network: The Next Wave in Distributed Processing?
<http://www.networkmagazine.com/article/NMG20020329S0009>

Un *Web Service* peut-être publié dans un registre situé à l'intérieur ou à l'extérieur d'un SI (Système d'information). Un *Web Service* peut être localisé en interrogeant le registre qui l'héberge. Une fois localisé, un *Web Service* peut être invoqué (même par un autre *Web Service*) en envoyant une requête appropriée.

- Capacité d'interagir avec d'autres composantes logicielles via des éléments XML et utilisant des protocoles de l'Internet

L'une des bases des *Web Services* est l'utilisation de protocoles standards de l'Internet tels que HTTP (*Hypertext Transfer Protocol*, le protocole du Web), SMTP (*Simple Mail Transfer Protocol*, le protocole du courriel électronique) et XML. Les *Web Services* peuvent donc traverser les coupe-feu conventionnels sans problème.

Contrairement à une page Web ou à une application de bureautique, les *Web Services* ne sont pas destinés à une interaction humaine directe. Ils sont plutôt conçus pour être utilisés par d'autres logiciels.

- Composante logicielle légèrement couplée à interaction dynamique

Par légèrement couplée, on veut dire que le *Web Service* et le programme (le consommateur de *Web Service*) qui l'invoque peuvent être modifiés indépendamment l'un de l'autre. Cela veut aussi dire que, contrairement à une composante logicielle qui serait fortement couplée, il n'est pas nécessaire de connaître la machine, le langage, le système d'exploitation ou tout autre détail, pour qu'une communication puisse avoir lieu. Cela offre une flexibilité qui permet aux entreprises d'éviter les coûts engendrés par l'intégration via des communications fortement couplées, telles que requises par l'EDI, par exemple.

Par interaction dynamique, on signifie que le consommateur de *Web Services* peut localiser et invoquer ce dernier au moment de l'exécution du programme sans avoir à programmer cette habileté à l'avance. Cela présuppose aussi que l'on peut modifier le *Web Service* sans avoir à modifier le logiciel de chacun des utilisateurs potentiels.

Tous ces éléments permettent de mieux comprendre ce que sont les *Web Services*. Chacun de ces éléments exprime une facette de ce qu'il est maintenant convenu d'appeler *Web Services*. Une définition globale d'un *Web Service* (au niveau technologique) est donc :

Une application logicielle, légèrement couplée, à interaction dynamique, identifiée par un URI, pouvant interagir avec d'autres composantes logicielles et dont les interfaces et liaisons ont la capacité d'être publiées, localisées et invoquées via XML et l'utilisation des protocoles Internet communs. Ils sont les bases permettant de construire des systèmes distribués et ouverts sur Internet, utilisant des technologies indépendantes des plates-formes.

« Ils sont les bases permettant de construire des systèmes distribués et ouverts sur Internet, utilisant des technologies indépendantes des plates-formes. »

2.2 Les technologies des Web Services

XML, SOAP (*Simple Object Access Protocol*), WSDL (*Web Services Description Language*) et UDDI (*Universal Description, Discovery and Integration*) sont les technologies dominantes des *Web Services*. Une pléthore d'autres technologies viendront, au fil du temps, garnir l'architecture des *Web Services*. Sans entrer dans tout les détails techniques, nous vous exposons ici les grandes lignes de chacune de ces technologies et de celles, qui à court et moyen terme, risquent de se positionner à la suite.

Fondamentalement, SOAP, WSDL et UDDI sont des technologies issues de l'intérêt parmi des membres de la communauté Internet à développer un mécanisme pour échanger des documents XML sur le Web entre systèmes d'information.

Dans cette discussion des technologies, il est aussi primordial de présenter ebXML qui se veut un cadre de référence pour l'utilisation de technologies basées sur XML dans le contexte du commerce électronique.

2.2.1 XML (eXtensible Markup Language)

XML est une famille de technologies développées au sein du W3C (*World Wide Web Consortium*). XML est née de la tentative de mettre SGML¹⁰ sur le Web. La première spécification de XML est apparue en février 1998 et se concentre sur les données, contrairement à HTML (par exemple) qui focalise sur la présentation. XML permet donc de transformer Internet d'un univers d'information et de présentation de sites Web statiques à un univers Web programmable et dynamique, centré sur les données. XML est largement utilisé par les entreprises et supporté par les manufacturiers informatiques. Il est indépendant des plates-formes informatiques. Il est lisible par l'humain mais est destiné à être lu par la machine. Il est flexible en ce sens que vous pouvez définir d'autres langages à partir d'XML. XML permet aux données d'être universellement navigables. Cependant, XML a aussi la particularité d'être adapté aux utilisateurs, ce qui signifie que la composition de ses balises (*tag*) peut grandement différer d'une entreprise ou d'une industrie à une autre. Certains y voient une limitation aux potentialités d'interopérabilité d'XML tandis que d'autres soulignent plutôt la versatilité avec laquelle XML peut coller aux différents besoins des entreprises et à la complexité des processus d'affaires¹¹.

Cette versatilité a donné naissance à une multitude de standards XML. ZapThink en a dénombré au-delà de 450¹², organisé en quatre catégories : les spécifications XML de bases, les spécifications orientées message, les spécifications orientées document et

« XML permet donc de transformer Internet d'un univers d'information et de présentation de sites Web statiques à un univers Web programmable et dynamique, centré sur les données. [...] Il est lisible par l'humain mais est destiné à être lu par la machine. »

¹⁰ SoftQuad, Inc. *The SGML Primer. SoftQuad's Quick Reference Guide to the Essentials of the Standard: The SGML Needed for Reading a DTD and Marked-up Documents and Discussing them Reasonably*. Version 3.0, May 1991.

¹¹ ZapThink, LLC, *The "Pros and Cons" of XML*, coll. ZapThink research report, 2001, p. 35.

¹² ZapThink, LLC, *Poster Key XML specifications*, Zapthink Document IDZTS-G1101, mai 2002.

les vocabulaires de communauté. Selon cette typologie, les *Web Services* se retrouvent dans la section « spécifications orientées message ».

2.2.2 SOAP (*Simple Object Access Protocol*)

SOAP est un protocole de la famille XML servant à l'échange d'informations dans un environnement distribué et décentralisé. Il est considéré comme la technologie la plus importante des *Web Services*¹³. Le standard SOAP a été proposé au W3C par Microsoft, IBM, Lotus, DevelopMentor et Userland.

Le standard SOAP définit trois éléments composant un message : l'enveloppe, l'en-tête du message et le corps du message. L'enveloppe définit le cadre pour décrire ce qui est dans le message et comment le traiter. Les règles d'encodages sont placées dans l'en-tête et servent à exprimer et définir le mécanisme de représentation des données. Le corps du message permet de transmettre les requêtes et les réponses entre les systèmes.

Plus de 80 implantations de la spécification SOAP v1.1 ont été développées jusqu'à présent. Cela démontre déjà la simplicité, la popularité et la justesse de l'approche qu'il offre pour transporter des données sur le Web^{14,15}.

2.2.3 WSDL (*Web Services Description Language*)

WSDL est un langage permettant de décrire les *Web Services* et les données attendues par ces *Web Services*. L'utilité de WSDL est donc de décrire et publier le format et les protocoles d'un *Web Service* de manière homogène par l'utilisation de XML. Cela permettra au requérant et à l'émetteur d'un service de comprendre les données qui seront échangées.

On peut comparer WSDL aux langages de description requis par CORBA et DCOM. Cependant, WSDL est différent de ceux-ci en ce sens qu'il est tout aussi neutre du point de vue du protocole que du point de vue de l'implantation¹⁶. Cependant, malgré que WSDL soit neutre du point de vue du protocole, la majorité des implantations WSDL se font à partir de SOAP puisque les promoteurs de SOAP sont aussi les promoteurs de WSDL¹⁷. WSDL a été développé conjointement par IBM, Microsoft et Ariba et a été présenté pour analyse au W3C qui l'a accepté comme une notice.

« SOAP est un protocole de la famille XML servant à l'échange d'informations dans un environnement distribué et décentralisé. Il est considéré comme la technologie la plus importante des Web Services. »

¹³ Eric Newcomer, *Understanding Web Services : XML, WSDL, SOAP and UDDI*, éd. Addison-Wesley, 2002, p.111.

¹⁴ Eric Newcomer, *Understanding Web Services : XML, WSDL, SOAP and UDDI*, éd. Addison-Wesley, 2002, p.117.

¹⁵ SoapWare.Org : Implementations, <http://www.soapware.org/directory/4/implementations>

¹⁶ *Web Services*, http://www.omg.org/news/meetings/workshops/webservices_2002.htm Mark Ferreira, Chief Scientist, Talking Blocks *Contracts for Services: Needs and Nonsense!*

¹⁷ William L. Oellermann, Jr., *Architecting Web Services*, éd. Apress, 2001, p.606.

2.2.4 UDDI (*Universal Description, Discovery and Integration*)

UDDI définit les mécanismes permettant de répertorier des *Web Services*. Ce standard régit donc l'information relative à la publication, la découverte et l'utilisation d'un *Web Service*. En d'autres mots, UDDI détermine comment nous devons organiser l'information concernant une entreprise et les *Web Services* qu'elle offre à la communauté afin de permettre à cette communauté d'y avoir accès. En fait, UDDI définit un registre des *Web Services* sous un format XML. Ce registre peut être public, privé ou partagé.

Tout comme WSDL, UDDI est une création du trio IBM, Microsoft et Ariba. Au départ ils ont développé la spécification puis ont rassemblé quelques 300 entreprises sous le chapeau de l'organisation UDDI.org afin de continuer le développement et de légitimer leurs efforts. UDDI a été déposé à OASIS¹⁸ (*Organization for the Advancement of Structured Information Standards*) en juillet 2002¹⁹ afin de permettre à cet organisme de standardisation de parrainer la spécification et d'en assurer son développement technique de façon indépendante. OASIS a officialisé son implication en créant le *OASIS UDDI Specification Technical Committee* en août 2002.

En septembre 2002, il existait trois registres publics (aussi appelés UBR ou *Universal Business Registry*), avec chacun son nœud de registre d'affaires et son nœud de tests. Ils sont entretenus et offerts par IBM, Microsoft et SAP²⁰.

Les registres privés et partagés peuvent être logés soit à l'intérieur des coupe-feu, soit hors coupe-feu grâce à certains intermédiaires²¹. Ces différents registres pourront aussi interagir entre eux. Déjà, les propriétaires des registres publiques s'échangent des informations à des fins de redondance. Ils pourront aussi interagir avec d'autres registres privés ou partagés auxquels ils auront accès.

Une entreprise choisit le répertoire dans lequel elle désire déposer les informations pertinentes à ses *Web Services*. Pour pouvoir publier ou rechercher un *Web Service*, des messages SOAP ont été définis. Ainsi, SOAP peut être utilisé pour publier (*publisher API*) et localiser (*consumer API*) d'autres services SOAP. Les entreprises et individus utilisent ces interfaces définies avec SOAP, ou encore l'interface utilisateur fournie par le propriétaire du registre UDDI en question, afin de s'enregistrer ou d'accéder aux informations du registre.

« UDDI définit les mécanismes permettant de répertorier des Web Services. Ce standard régit donc l'information relative à la publication, la découverte et l'utilisation d'un Web Service. »

¹⁸ OASIS, <http://www.oasis-open.org/>.

¹⁹ *Web Services Body*, UDDI.org, *Transitions Work to OASIS Standards Consortium UDDI.org*, http://www.uddi.org/news/uddi_news_07_30_02.html.

²⁰ Registres publics: UDDI.org, <http://www.uddi.org/register.html> et UDDI-China.ORG - Universal Description, Discovery and Integration, <http://www.uddi-china.org/register/>.

²¹ Voir :

- ZDNet: Tech Update: eBusiness / Breathing new life into UDDI, <http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2872349,00.html>
- E2open | Global Collaboration Network, <http://www.e2open.com>
- Press Release - Software AG: Demo Version of UDDI Repository and Tools Now Available, <http://www.softwareag.com/corporat/news/august2001/UDDI.htm>

2.2.5 L'architecture ebXML

L'architecture ebXML vient de l'intérêt des membres de la communauté EDI à développer une manière plus efficace pour échanger des documents d'affaires en utilisant XML sur des connections Internet plutôt que sur des VAN (*Value Added Network*) dispendieux²². Cette architecture a été développée par OASIS et UN/CEFACT²³ (*United Nations Centre for Trade Facilitation and Electronic Business*). Pour développer cette architecture, les concepteurs sont partis des processus d'affaires pour en arriver à l'identification des besoins de communication au niveau de la couche transport. Ceci contraste avec l'approche de développement prônée par les instigateurs des *Web Services* qui sont parti des mécanismes de transport pour finalement en arriver à la problématique des processus d'affaires²⁴. Ainsi, ebXML et les *Web Services* se complètent bien.

L'architecture ebXML ayant à son origine l'EDI, très axé sur les processus d'affaires complexes des grandes entreprises, de nombreux critiques²⁵ soulèvent le fait que les besoins des grandes entreprises ne correspondent pas nécessairement à ceux des PME. Par ailleurs, la légitimité «politique» d'ebXML est sans doute l'argument qui milite le plus en sa faveur. Sous les chapeaux d'OASIS et d'UN/CEFACT, ebXML rejoint plus de 4500 participants représentant 2000 organisations sur tous les continents.

Conceptuellement, ebXML consiste en six principales spécifications²⁶ :

- une architecture technique, offrant un survol et un résumé détaillé de l'architecture complète. Il s'agit d'une infrastructure visant l'interopérabilité, tout comme les *Web Services*;
- les BPSS²⁷ (*Business Process Specification Schema*) et la description des méthodologies pour produire un BPSS, Un BPSS spécifie un processus d'affaires cadre. Ainsi, les différentes formes d'interactions définies dans le processus d'affaires assurent la chorégraphie des transactions d'affaires entre les collaborateurs. Le BPSS inclut donc des profils de collaboration d'affaires visant à permettre l'échange de message et de signaux entre les partenaires. Cet élément n'est pas inclus dans les *Web Services*, dans leur forme actuelle.
- Les CPP/A²⁸ (*Collaboration Protocol Profile / Agreement*), décrivant les habiletés des parties à conclure une transaction d'affaires sur Internet.

« L'architecture ebXML vient de l'intérêt des membres de la communauté EDI à développer une manière plus efficace pour échanger des documents d'affaires en utilisant XML sur des connections Internet plutôt que sur des VAN (Value Added Network) dispendieux »

²² Eric Newcomer, *Understanding Web Services : XML, WSDL, SOAP and UDDI*, éd. Addison-Wesley, 2002, p.189.

²³ United Nations Centre for Trade Facilitation and Electronic Business, <http://www.unece.org/cefact/>

²⁴ Eric Newcomer, *Understanding Web Services : XML, WSDL, SOAP and UDDI*, éd. Addison-Wesley, 2002, p.200.

²⁵ Voir en particulier : ebXML and SMEs, <http://www.rawlinseconsulting.com/ebXML/ebXML4.html>

²⁶ Traduit librement de : Eric Newcomer, *Understanding Web Services : XML, WSDL, SOAP and UDDI*, éd. Addison-Wesley, 2002, p.203-204.

²⁷ <http://www.ebxml.org/specs/ebBPSS.pdf>

²⁸ <http://www.ebxml.org/specs/ebCCP.pdf>

Notamment, le CPP sert à enregistrer et accéder à l'information relative aux modes de transports disponibles (c.-à-d., SMTP ou HTTP), aux exigences de sécurité (encryption, non-répudiation et signature digitale) et aux processus d'affaires disponibles pour exécuter la transaction. On peut comparer un CPP à un document WSDL. Cependant, le CPP fournit beaucoup plus d'information. Le CPA²⁹ sert à décrire l'entente entre deux parties sur les spécificités techniques définissant les caractéristiques, processus et services de la relation d'affaire électronique entre les deux parties. Deux parties créeront donc un CPA en se basant sur les capacités technologiques communes déjà identifiées dans leur CPP respectif.

- Un service de registre et de dépôt de données servant à entreposer et récupérer les spécifications et les modèles de processus d'affaires, l'identité des partenaires commerciaux et les exigences technologiques nécessaires à une transaction. Ce registre offre des fonctionnalités similaires à celles offertes par UDDI.
- Un service de messagerie décrivant comment un document sera transportée sur le réseau, lorsque l'architecture ebXML est utilisée. À l'origine, les concepteurs de ebXML ont défini le MSS³⁰ (*Message Service Specification*) afin de réaliser ce service de messagerie. Cependant, en mars 2001, les responsables de ebXML ont incorporé le protocole SOAP à leur spécification, créant une version améliorée de SOAP, soit SOAP avec attachement.
- Des composantes élémentaires fournissant une fondation sémantique pour les messages échangés. Ces composantes sont définies indépendamment du contexte d'utilisation. Ceci permet de réutiliser la même composante dans des contextes différents.

3 Situation actuelle des *Web Services* dans les entreprises

Déjà, plusieurs entreprises américaines ont utilisées les *Web Services* pour offrir des services à valeur ajoutée. Selon Michael Vizard de InfoWorld³¹, les impacts des *Web Services* se font sentir à court terme. Par exemple, il cite le cas de Merrill Lynch qui a

« Le CPA sert à décrire l'entente entre deux parties sur les spécificités techniques définissant les caractéristiques, processus et services de la relation d'affaire électronique entre les deux parties. »

²⁹ Traduit librement de : Kotok et. al., *ebXML: The new global standard for doing business over the internet*, éd. New Riders, sept. 2001, ISBN 0-7357-1117-8, p. 251.

³⁰ <http://www.ebxml.org/specs/ebMS2.pdf>

³¹ *Web Services* are delivering savings, <http://www1.infoworld.com/cgi-bin/fixup.pl?story=http://www.infoworld.com/articles/op/xml/02/08/19/020819opnoise.xml&dctag=webservices>.

réussi un projet d'intégration à un coût de \$30 000 plutôt que les \$800 000 initialement prévus, grâce à l'utilisation des *Web Services*.

Dans un autre projet³², Merrill Lynch a réussi à intégrer des données provenant de près de 2000 différents SI de support à la clientèle. Dans la première phase de ce projet, Merrill Lynch a développé un système d'analyse de portfolio à partir des données sur la clientèle, les produits et sur les marchés en temps réel. La deuxième phase de ce projet vise à rendre ces données accessibles à partir de téléphones traditionnels et mobiles, ainsi qu'à partir d'assistants numériques personnels (*Personal Digital Assistants*).

Goldman Sachs (cité dans Hagel III³³) indique que les économies résultant des initiatives *Web Services* de la chaîne d'approvisionnement de GM permettront à cette dernière d'économiser \$1000/véhicule en coûts d'opérations.

Hagel III mentionne aussi que Citibank³⁴ a développé CitiConnect à l'aide des *Web Services*. Ce service permet le traitement des paiements électroniques pour les places de marché. Grâce à cette initiative, le temps de règlement des comptes entre les acheteurs et les vendeurs a été réduit de 20 à 40% et les coûts associés au règlement de la transaction autant pour les acheteurs que les vendeurs ont été réduits de 50 à 60%.

Chez Dell³⁵, on se sert des *Web Services* afin de rendre disponible à toutes les 24 heures les horaires de production, permettant à ses fournisseurs à leur tour d'ajuster leurs livraisons. Cette initiative a permis à Dell de réduire le temps de détention sur stock de 26-30 heures à 3-5 heures, soit une réduction de 80%.

Un autre exemple³⁶ d'utilisation des *Web Services* est le *Colorado Department of Agriculture* qui se sert des *Web Services* afin de publier les données de repérage des chevreuils et des cerfs. Cet organisme public avait des données qui résidaient sur une base de données Macintosh, des résultats de tests de laboratoires sur ordinateur central et d'autres fichiers connexes disponibles sur différentes plateformes. Suite à l'effort d'intégration à l'aide des *Web Services*, les données sont maintenant disponibles en temps réel plutôt qu'avec 3 à 4 semaines de délai.

Plusieurs autres entreprises ont aussi eu des résultats très probants en utilisant cette technologie : AAA³⁷, Amazon³⁸, Continental Airlines³⁹, Dell⁴⁰, DuPont Performance

« Merrill Lynch a réussi un projet d'intégration à un coût de \$30 000 plutôt que les \$800 000 initialement prévus, grâce à l'utilisation des *Web Services*. »

³² Dyck, Timothy, *Web Services impact*, eWeek, Ziff Davis Media, Vol. 19, numéro 37.

³³ Ce cas est discuté dans l'article de Hagel III, Brown, [Break on Through to the Other Side: A Missing Link in Redefining the Enterprise](http://johnhagel.com/consulting.html#focus), 2002 johnhagel.com: Where Business meets IT, <http://johnhagel.com/consulting.html#focus>.

³⁴ Ce cas est discuté dans l'article de Hagel III, Brown, [Break on Through to the Other Side: A Missing Link in Redefining the Enterprise](http://johnhagel.com/consulting.html#focus), 2002 johnhagel.com: Where Business meets IT, <http://johnhagel.com/consulting.html#focus>.

³⁵ op.cit.

³⁶ Dyck, Timothy, article: *Web Services impact*, eWeek, Ziff Davis Media, Vol. 19, numéro 37.

³⁷ AAA launches Web airline reservation system - Computerworld, <http://www.computerworld.com/developmenttopics/websitemgmt/story/0,10801,73426,00.html>.

Coating⁴¹, Dollar Rent a Car⁴², ebay⁴³, Fedex, General Motors^{44,45}, Google⁴⁶, Home Depot⁴⁷, JP Morgan Chase⁴⁸, Merrill Lynch^{49,50}, Nasdaq^{51,52}, UPS⁵³ et Zagat⁵⁴.

Parmi ces entreprises, certaines ont commencé à exposer des *Web Services* hors des frontières de l'entreprise, pour leurs différents clients et partenaires. Cela veut dire que les *Web Services* changent déjà l'horizon B2B.

Contrairement à ce qu'on aurait pu croire, ce ne sont donc pas les PME désirant réduire les coûts liés à l'EAI (*Enterprise Application Integration*) qui sont les premiers utilisateurs des *Web Services*⁵⁵ mais plutôt les grosses entreprises qui s'en

« ... les Web Services
changent déjà l'horizon
B2B. »

³⁸ Amazon.com *Web Services*, http://associates.amazon.com/exec/panama/associates/ntg/browse/-/1067662/ref=gw_hp_ls_1_3/086-7154800-3213037.

³⁹ Continental Airlines, <http://msdn.microsoft.com/vstudio/productinfo/casestudies/continental/default.asp>.

⁴⁰ Ce cas est discuté dans l'article de Hagel III, Brown, *Break on Through to the Other Side: A Missing Link in Redefining the Enterprise*, 2002 johnhagel.com: Where Business meets IT, <http://johnhagel.com/consulting.html#focus>.

⁴¹ Bowstreet - DuPont Performance Coatings selects Bowstreet to automate custom portals for thousands of auto body shops, http://www.bowstreet.com/newsevents/pressreleases/112601_dupont_selects_bowstreet.html.

⁴² Microsoft Case Studies: Dollar Rent A Car Systems, Inc., <http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=11626>.

⁴³ Cover Pages: eBay Inc. and Microsoft Announce SOAP-based XML *Web Services* for Online E-Commerce., <http://xml.coverpages.org/ni2001-03-15-d.html>.

⁴⁴ InformationWeek > *Web Services* > GM Hopes *Web Services* Turn The Key To Data Access > March 15, 2002, <http://www.informationweek.com/story/IWK20020315S0027>.

⁴⁵ Ce cas est aussi discuté dans l'article de Hagel III, Brown, *Break on Through to the Other Side: A Missing Link in Redefining the Enterprise*, 2002 johnhagel.com: Where Business meets IT, <http://johnhagel.com/consulting.html#focus>.

⁴⁶ Google Web APIs - Home, <http://www.google.com/apis/>.

⁴⁷ The Home Depot's latest project: XML, *Web Services*, http://www.nwfusion.com/news/2002/129295_01-21-2002.html.

⁴⁸ IBM e-business: jStart program: Case studies: J.P. Morgan Chase & Co., <http://www-3.ibm.com/software/ebusiness/jstart/casestudies/jpmorganchase.html>.

⁴⁹ Merrill Lynch charges into *Web Services* - Computerworld, <http://www.computerworld.com/developmenttopics/development/webdev/story/0,10801,70954,00.html>.

⁵⁰ Ce cas est aussi discuté dans l'article de Hagel III, Brown, *Break on Through to the Other Side: A Missing Link in Redefining the Enterprise*, 2002 johnhagel.com: Where Business meets IT, <http://johnhagel.com/consulting.html#focus>.

⁵¹ www.xmethods.net, [http://www.xmethods.com/ve2/ViewListing.po;jsessionid=13WVg7N9X3d6QpsCjEz7LXu7\(QhxieSRM\)?serviceid=110765](http://www.xmethods.com/ve2/ViewListing.po;jsessionid=13WVg7N9X3d6QpsCjEz7LXu7(QhxieSRM)?serviceid=110765)

⁵² Microsoft Case Studies: Nasdaq.com, <http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=11485>.

⁵³ Shipping & Receiving *Web Services* (service), <http://www.remotemethods.com/home/business/shipping>.

⁵⁴ Solve Real Business Problems, <http://msdn.microsoft.com/vstudio/productinfo/solve.asp>.

⁵⁵ Scott Durschlag, *Beyond the hype... The reality of early web service adoption*, Focus Security, mars 2002, http://www.grandcentral.com/pdf/beyond_the_hype-durchslag.pdf.

servent afin de rationaliser leurs dépenses, que se soit au niveau de la chaîne d'approvisionnement, de la chaîne de gestion de la demande, des marchés privés, ou de l'intégration des applications et des fournisseurs de services *Web*.

3.1 Perception des Web Services par les gestionnaires TI

Qu'en est-il des autres entreprises? Où en sont leurs différentes implantations des *Web Services* et quelle est la perception qu'ont les gestionnaires TI de ce nouveau paradigme informatique? Selon Gartner⁵⁶, la perception qu'ont les gestionnaires TI des *Web Services* devrait varier selon un cycle qui se module sur les attentes irréalistes générées par le tapage médiatique (fig. 1). Gartner identifie ainsi deux vagues de la technologie des *Web Services*. La première étant celle générée par le Espeak de Hewlett Packard et débutant aussitôt qu'en 1998. La deuxième est celle de 2001, correspondant à l'explosion de l'offre de la technologie des *Web Services*. C'est cette dernière vague qui devrait subir les contrecoups de la désillusion vers la fin de 2003.

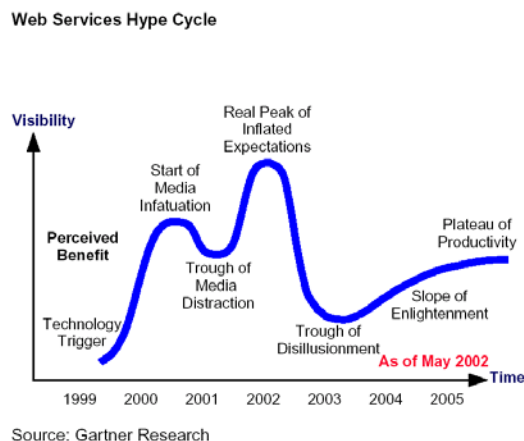


Figure 1⁵⁷

Incidemment, les gestionnaires qui donnent leur avis aux différents organismes qui les sonde, seront affectés par tout ce tapage publicitaire. Si on admet la théorie de Gartner, cette recherche, ainsi que le sondage de Infoworld⁵⁸ tous deux réalisés en 2002, devaient être passablement biaisés. Néanmoins, Infoworld illustre de façon évidente la perception actuelle de son réseau de CTO (*Chief Technology Officer*) sondé. On y apprend (fig. 2) :

⁵⁶ Article de Gartner Research, [Explaining Web Services' Apparent Contradictions](#), David Smith 4 juin 2002.

⁵⁷ Article de Gartner Research, [Explaining Web Services' Apparent Contradictions](#), David Smith 4 juin 2002.

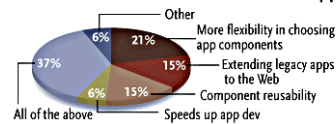
⁵⁸ App dev on the *Web Services* path, <http://www.infoworld.com/articles/fe/xml/02/06/10/020610feappdev.xml>.

« Contrairement à ce qu'on aurait pu croire, ce ne sont donc pas les PME [...] qui sont les premiers utilisateurs des Web Services mais plutôt les grosses entreprises qui s'en servent afin de rationaliser leurs dépenses, que se soit au niveau de la chaîne d'approvisionnement, de la chaîne de gestion de la demande, des marchés privés, ou de l'intégration des applications et des fournisseurs de services *Web*. »

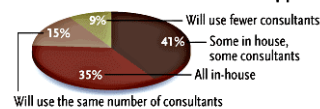
« ... 37% des CTO interrogé identifient déjà plusieurs des bénéfices promis par les *Web Services* »

- que 37% des CTO interrogé identifient déjà plusieurs des bénéfices promis par les *Web Services*;
- qu'ils auront besoin de consultants externes dans 41% des cas et qu'un autre 9% croient que moins de consultants seront nécessaires;
- que 74% utilisent déjà les *Web Services* pour l'intégration d'applications;
- que 68% croient que tous leurs systèmes seront intégrés et disponibles sous formes de *Web Services* d'ici 5 ans.

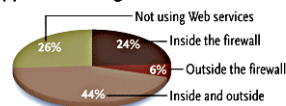
What are the benefits of Web services app dev?



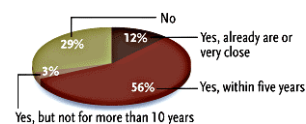
How will Web services affect your use of in-house staff or consultants for app dev?



Where are you using Web services for application integration?



Do you foresee all your systems being Web services-enabled and available as Web services?



SOURCE: INFOWORLD CTO NETWORK SURVEY

Figure 2⁵⁹

Effectivement, ces chiffres donnent à penser que les *Web Services* jouissent de ce momentum que Gartner appelle le « *Real peak of inflated expectations* ». N'oublions tout de même pas que Merrill Lynch et les autres innovateurs cités plus haut apprécient déjà les avantages de ces technologies.

Selon Deloitte & Touche⁶⁰ et IDC⁶¹, l'évolution des *Web Services* devrait se faire en trois phases (fig. 3). Une première phase concerne l'intégration des applications disparates d'une entreprise et se fera à l'intérieur du coupe-feu. Cette phase permettra

« Une première phase concerne l'intégration des applications disparates d'une entreprise et se fera à l'intérieur du coupe-feu. Cette phase permettra aux entreprises d'optimiser leurs processus d'affaires, en mettant l'accent sur la relation client plutôt que sur les problèmes d'intégration. »

⁵⁹ App dev on the *Web Services* path, <http://www.infoworld.com/articles/fe/xml/02/06/10/020610feappdev.xml>.

⁶⁰ Deloitte & Touche, The Blue Paper, *Web Services: The next evolution in software*, 2002, http://www.allidex.com/DTCF_Web_Services_Research_Report.pdf.

⁶¹ Altering app dev, <http://www.infoworld.com/articles/fe/xml/02/06/10/020610feinfostat.xml>.

aux entreprises d'optimiser leurs processus d'affaires, en mettant l'emphase sur la relation client plutôt que sur les problèmes d'intégration. Les entreprises utiliseront les emballages XML et SOAP pour encapsuler leurs applications essentielles aux processus d'affaires fondamentaux, afin de diminuer les coûts associés aux efforts d'intégration d'applications d'entreprises.

Web services progression	
2002-2004	Within the firewall <ul style="list-style-type: none"> • Simplified application integration • Increased developer productivity
2004-2006	Contained external users <ul style="list-style-type: none"> • Simplified business partner connectivity • Richer application functionality • Subscription-based services
2006-2008	Fully dynamic search and use <ul style="list-style-type: none"> • Casual/ad-hoc use of services • New business models possible • Commoditization of software • Pervasive use in nontraditional devices

SOURCE: IDC

Figure 3⁶²

La deuxième phase, tout comme la première, concerne l'intégration des applications. Cette fois-ci, cette intégration se fera hors du coupe-feu, l'une des limites actuelles des *Web Services* étant la sécurité (cf. sect. 5.3). Plusieurs solutions et protocoles sont déjà proposés pour remédier à cette situation et à mesure que ces solutions se concrétiseront, de plus en plus d'entreprises utiliseront les *Web Services* hors de leurs réseaux privés.

La troisième phase permettra l'éclosion des applications *Web Services*. De nouveaux modèles d'affaires seront possibles, les entreprises pourront potentiellement réutiliser des applications développées par d'autres entreprises et les logiciels deviendront une commodité. Cependant, Deloitte & Touche fait aussi remarquer que les processus d'affaires fondamentaux reliés à la compétitivité d'affaire d'une entreprise, ne seront certainement pas disséminés hors du cercle restreint des partenaires et clients de confiance.

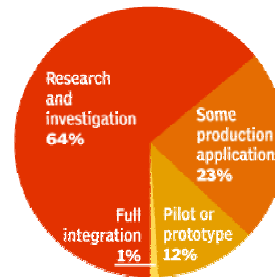
Une autre étude sur le sujet a été publiée dans PC Magazine⁶³. Cette étude a été réalisée par Giga Information Group auprès de décideurs majeurs en TI, présents à une de leur conférence. On y apprend que malgré le climat économique actuel, les répondants misent sur les *Web Services*, 36% étant déjà en projet pilote ou en phase de production d'applications utilisant les *Web Services* (fig. 4).

« ... malgré le climat économique actuel, les répondants misent sur les Web Services, 36% étant déjà en projet pilote ou en phase de production d'applications utilisant les Web Services. »

⁶² Altering app dev, <http://www.infoworld.com/articles/fe/xml/02/06/10/020610feinfoestat.xml>.

⁶³ iBiz Stats (v21n12), <http://www.pcmag.com/article2/0,4149,4408,00.asp>.

How far along is your company in using *Web Services* technologies such as SOAP, WDSL, and UDDI?

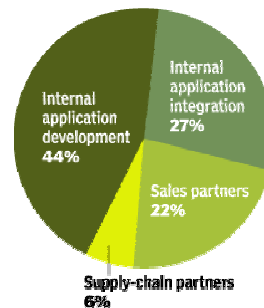


77 responses. Source: Giga Information Group, 2002.

Figure 4⁶⁴

Nous y apprenons aussi que 71% des répondants favorisent l'utilisation des *Web Services* pour des projets internes (fig. 5). Ceci est conforme aux prédictions de Gartner et IDC quant à l'utilisation des *Web Services*.

What is the primary target for your *Web Services* development?

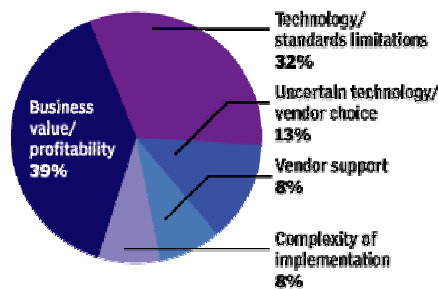


63 responses. Source: Giga Information Group, 2002.

Figure 5⁶⁵

Finalement, l'incertitude la plus répandue chez ces dirigeants (39%) est associée à la perception qu'ils ont de la valeur d'affaires des *Web Services* (fig. 6).

What is the biggest challenge facing your *Web Services* strategy?



76 responses. Source: Giga Information Group, 2002.

Figure 6⁶⁶

Une autre question d'importance pour les gestionnaires des TI et affectant directement le retour sur investissement est la question des coûts comparatifs d'intégration en utilisant les *Web Services* plutôt que d'autres méthodes

« ... l'incertitude la plus répandue chez ces dirigeants (39%) est associée à la perception qu'ils ont de la valeur d'affaires des *Web Services*. »

⁶⁴ iBiz Stats (v21n12), <http://www.pcmag.com/article2/0,4149,4408,00.asp>.

⁶⁵ iBiz Stats (v21n12), <http://www.pcmag.com/article2/0,4149,4408,00.asp>.

⁶⁶ iBiz Stats (v21n12), <http://www.pcmag.com/article2/0,4149,4408,00.asp>.

traditionnelles d'intégration. La firme Zapthink, dans une étude récente⁶⁷, fait ressortir que l'utilisation des *Web Services* à des fins d'intégration (dans une optique d'intégration orientée service) est nettement plus rentable que les approches d'intégration traditionnelles (EAI et B2BI - *Business-to-Business Integration*), que les approches sur mesure (*custom made*) ou que la simple utilisation d'API *Web Services* à des fins d'intégration. Ici, nous tenons à spécifier la différence entre une approche d'intégration orientée service utilisant des *Web Services*, d'une part, et une intégration n'utilisant que des API *Web Services*, d'autre part. Dans le cas de l'intégration orientée service utilisant les *Web Services* ou SOI (*Service Oriented Integration*), l'intégration est basée sur la définition de services, ce qui sous-tend une analyse profonde des processus d'affaires et la mise sur pied d'une architecture d'intégration légèrement couplée. Cette architecture expose directement les processus d'affaires sous forme de service et ce à différents niveaux de granularité (la granularité est le niveau de fragmentation d'une unité donnée, d'un service). Dans le cas d'API *Web Services*, l'intégration se limite à définir des interfaces utilisant les *Web Services*, de la même manière qu'on utiliserait CORBA ou DCOM.

Ainsi, selon ZapThink (fig. 7), les coûts initiaux de l'effort de réingénierie des processus requis par le SOI seront plus élevés. Par contre, les coûts d'entretien et de changement de cette solution seront beaucoup moins importants. Les coûts liés à l'utilisation d'API *Web Services* (*Web Services « Adapters »*), quant à eux, ressembleront aux coûts d'utilisation des approches traditionnelles d'intégration.

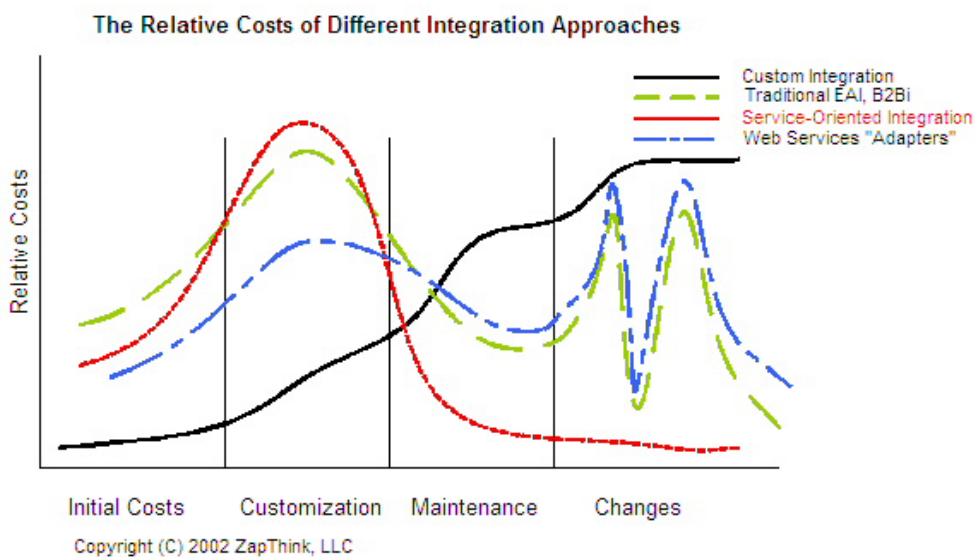


Figure 7⁶⁸

« ... l'utilisation des Web Services à des fins d'intégration est nettement plus rentable que les approches d'intégration traditionnelles, que les approches sur mesure ou que la simple utilisation d'API Web Services à des fins d'intégration. »

« ... les coûts initiaux de l'effort de réingénierie des processus requis par le SOI seront plus élevés. Par contre, les coûts d'entretien et de changement de cette solution seront beaucoup moins importants. »

⁶⁷ Schmelzer et.al. article: *Understanding the Real Costs of Integration*, ZapFlash, Zapthink Research, oct. 2003.

⁶⁸ Schmelzer et.al. article: *Understanding the Real Costs of Integration*, ZapFlash, Zapthink Research, oct. 2003.

4 Les avantages des *Web Services*

L'idée fondamentale derrière les *Web Services* est de morceler les applications et les processus d'affaires en morceaux réutilisables appelés « Service » de sorte que chacun de ces segments effectue une tâche distincte. Ces services peuvent alors servir à l'intérieur et à l'extérieur de l'entreprise, facilitant l'interopérabilité entre tous ces services. De par leur nature, les *Web Services*⁶⁹ :

- permettent à des portions de logiciels écrits dans différents langages, ou évoluant sur différents systèmes d'exploitation, de communiquer entre elles facilement et à peu de frais;
- permettent à des applications supportant différents processus d'une organisation ou de différentes organisations, de communiquer entre elles et/ou d'échanger des données facilement et à peu de frais.

Les *Web Services* devraient permettre à de nouveaux écosystèmes d'affaires de voir le jour⁷⁰. Présentement, les systèmes ERP (*Entreprise Resource Planning*) ne partagent pas de logique d'affaire hors du coupe-feu d'entreprise, les systèmes EDI ne fonctionnent pas avec les consommateurs et nombres de petites entreprises, et les systèmes SCM (*Supply Chain Management*) tout comme les places de marchés, sont souvent emprisonnés dans des langages XML propriétaires ou encore réservés à une seule industrie. Les *Web Services* devraient permettre la création de concentrateurs de réseau de processus d'affaires. Plusieurs entreprises se positionnent déjà dans ce nouveau marché : Asera, BowStreet, Clarus, CommerceOne, Covast, Cyclone Commerce, eConnections, Excelon, Fiorano, Fujitsu/Glovia, Hubspan, I2, IBM, Iona, iPlanet, Manugistics, MatrixOne, ModelN, Oracle, SAP-SAPMarkets, Ventro, VerticalNet, Viacore et Viquity⁷¹.

Plus spécifiquement, les *Web Services* devraient permettre aux entreprises⁷² de :

- donner aux clients un accès direct à l'information, aux données et aux fonctionnalités dont ils ont besoin pour interagir avec une entreprise;
- donner aux partenaires d'une entreprise un accès direct à la fonctionnalité dont ils ont besoin pour mieux servir les clients qu'ils ont en commun avec cette entreprise;

« Les Web Services
devraient permettre à de
nouveaux écosystèmes
d'affaires de voir le jour. »

⁶⁹ Adaptation de O'Reilly Network: *Web Services - An Executive Summary* [Apr. 12, 2002], <http://www.oreillynet.com/pub/a/webservices/2002/04/12/execreport.html>.

⁷⁰ Voir article : *The Aberdeen Group, The next B-toB Gestalt: Business Process Networks*, Market Viewpoint, Vol. 15, no.2, March 2002.

⁷¹ Aberdeen Group, *The next B-toB Gestalt: Business Process Networks*, Market Viewpoint, Vol. 15, no.2, March 2002.

⁷² Points tirés et traduits librement de : *Patricia Seybold Group, An executive's guide to web services, How to optimize Web Services investments to improve your customer experience*, Patricia Seybold Group's executive series, 2002, p. 13.

- donner aux fournisseurs d'une entreprise un accès direct à l'information et la fonctionnalité dont ils ont besoin pour leur permettre d'ajuster les inventaires selon le modèle « juste à temps »;
- intégrer des applications de bout en bout, de manière abordable, facile à implanter autant hors des frontières de l'entreprise qu'à l'intérieur de celles-ci;
- avoir des équipes de développement travaillant indépendamment et efficacement sur des systèmes qui interagiront, parce que ces équipes travailleront à développer des interfaces communes plutôt que d'avoir à synchroniser les processus.

Les *Web Services* offrent donc aux entreprises la flexibilité de réponse et d'anticipation des besoins changeant des clients, la rationalisation des infrastructures logicielles et la flexibilité d'interaction et de configuration des alliances externes avec les partenaires et fournisseurs.

4.1 L'explosion du nombre de partenaires

Une autre problématique à laquelle font face les entreprises se tournant vers Internet est le nombre croissant de partenaires potentiels⁷³. Cette croissance est souvent liée à une augmentation des coûts liés à l'élaboration et le déploiement des interfaces entre les systèmes d'information de ces partenaires. En particulier, trois défis se posent :

- Distribution des centres de contrôles : Les entreprises peuvent dicter l'utilisation d'une plate-forme homogène à l'intérieur de leurs frontières. Ils peuvent même obliger certains de leurs fournisseurs à s'adapter à cette plate-forme s'ils ont une position dominante déterminante. Cependant, lorsque le nombre et la diversité des partenaires augmentent, il devient difficile de maintenir un seul centre de contrôle;
- Diversité des plates-formes technologiques : Sans un centre de contrôle unique, les entreprises se battent continuellement avec la diversité croissante des plates-formes qu'ils ont à brancher. Ces branchements se doivent aussi d'être abordables et réalisables pour les PME qui doivent aussi supporter les coûts de ses branchements;
- L'environnement dynamique : Dans un monde économique en mouvance perpétuelle, les entreprises se doivent d'être capable d'intégrer les nouveaux partenaires à leurs systèmes informatiques et ce, de façon efficace, rapide et économique. Ils doivent aussi avoir la flexibilité d'abandonner certaines alliances d'affaires sans avoir à radier de leurs bilans des dépenses et investissements technologiques.

« Les Web Services offrent donc aux entreprises la flexibilité de réponse et d'anticipation des besoins changeant des clients, la rationalisation des infrastructures logicielles et la flexibilité d'interaction et de configuration des alliances externes avec les partenaires et fournisseurs. »

⁷³ Traduction libre de : Hagel III, John, **Out of the box: Strategies for achieving profits today and growth tomorrow through Web Services**, Harvard Business school press, 2002, p.22 à 25.

En réponse à ces défis, les *Web Services* offrent les solutions suivantes :

- La simplicité : Les *Web Services* réduisent la complexité des branchements tout en rendant la tâche plus facile aux nouveaux participants. Cela se fait en ne créant la fonctionnalité qu'une seule fois plutôt qu'en obligeant tous les participants à reproduire la fonctionnalité à chacun des bouts (comme avec l'architecture client/serveur);
- Composante logicielle légèrement couplée : L'architecture modulaire des *Web Services*, combinée au faible couplage des interfaces associées, permet l'utilisation et la réutilisation de services qui peuvent facilement être recombinaés à différents autres modules;
- Hétérogénéité : Les *Web Services* permettent d'ignorer l'hétérogénéité entre les différentes applications et modules. En effet, ils décrivent comment transmettre un message (standardisé) entre deux applications, sans imposer comment construire ce message;
- Ouverture : Les *Web Services* permettent de réduire les inquiétudes liées aux différents «*lock-in*» que les entreprises subissent des fournisseurs informatiques. Ils permettent aussi de tirer une valeur économique supplémentaire des infrastructures informatiques existantes et des plateformes ouvertes tel que l'Internet.

4.2 Impacts futurs des Web Services

Slywotsky et. Morisson⁷⁴ font bien ressortir l'importance pour le succès d'une entreprise de l'innovation technologique et de son incorporation à sa propre chaîne de valeur et son modèle d'affaire. Les *Web Services* devraient être un élément technologique primordial à ce chapitre, facilitant la découverte de nouvelles chaînes de valeurs encore insoupçonnées. Ainsi, nous croyons que les *Web Services* permettront de développer plus rapidement et à meilleur coûts quelques une des innovations suivantes :

- les tableaux de sélection (*choiceboard*), tel que chez Dell;
- liens électroniques entre l'entreprise, ses clients et fournisseurs;
- la formation en ligne;
- l'embauche en ligne;
- la création de communautés de clients et fournisseurs en ligne tel que chez Cisco;
- la multiplication des canaux de distributions;

« Les Web Services devraient être un élément technologique primordial à ce chapitre, facilitant la découverte de nouvelles chaînes de valeurs encore insoupçonnées. »

⁷⁴ Slywotsky, Adrian J., et. al., How digital is your business? éd. Crown Business, 2000.

- la personnalisation des différentes interfaces (client, employé, gestionnaire, fournisseur);
- les outils de diagnostic à distance, tel que chez GE.

Les *Web Services* devraient supporter les processus d'affaires légèrement couplés⁷⁵. En utilisant ce genre de processus, une entreprise peut, selon les besoins, établir dynamiquement des liens avec divers partenaires. La figure 8 montre comment ces processus d'affaires légèrement couplés diffèrent de l'approche traditionnelle de gestion des processus d'affaires fortement couplés. Il en ressort que les processus d'affaires légèrement couplés permettront de créer des réseaux étendus de processus d'affaires qui seront hautement spécialisés, dynamiques et hautement profitables. Bien qu'il soit concevable d'opérationnaliser cette approche sans l'aide des *Web Services*, cela s'avérera difficile pour les entreprises ayant des processus d'affaires complexes. Cette approche illustre de manière éloquente les liens qui existent entre le potentiel technologique et les perceptions que l'on se fait des modèles d'affaires. L'innovation en gestion n'a pas besoin de technologies pour exister. Cependant, il existe un lien entre la capacité de livrer ou d'opérationnaliser l'innovation en gestion et les contraintes technologiques des TI auxquelles ces innovations devront s'adapter. Les *Web Services* permettent de repousser la limite de ces contraintes technologiques; ils permettront à de nouvelles approches, tels que celle des processus d'affaires légèrement couplés, de voir le jour.

« il existe un lien entre la capacité de livrer ou d'opérationnaliser l'innovation en gestion et les contraintes technologiques des TI auxquelles ces innovations devront s'adapter. Les Web Services permettent de repousser la limite de ces contraintes technologiques; ils permettront à de nouvelles approches, tels que celle des processus d'affaires légèrement couplés, de voir le jour. »

Contrasting Approaches to Business Process Management

	Hard wired business processes	Loosely coupled business processes
	From	To
Roles	Controller Limited, all-purpose service providers	Orchestrator Increasingly specialized service providers
Rules	Management of micro-activities Instructions (push) Full information transparency	Management of macro-entities Incentives (pull) Selective information visibility
Renewal	Infrequent benchmarking Infrequent reengineering (every 5-10 years)	Continuous benchmarking Dynamic reconfiguration
Rewards	Experience effects Diminishing returns	Growing and continuous specialization Increasing returns

Figure 8⁷⁶

⁷⁵ Hagel III et. al., Orchestrating Loosely Coupled Business Processes: The Secret to Successful Collaboration, 2002.

⁷⁶ Hagel III et. al., Orchestrating Loosely Coupled Business Processes: The Secret to Successful Collaboration, 2002, p. 4.

Pour terminer, on notera que la nature flexible des *Web Services* fera en sorte qu'ils pourront être utilisés sur une foule de support (voitures, PDA, téléphone, etc.).

5 Les limites des *Web Services*

Les *Web Services* ne sont pas une panacée⁷⁷ : la technologie se doit d'être au service d'objectifs d'affaires et non l'inverse. De plus, comme tout effort technologique, l'implantation des *Web Services* requièrent des objectifs clairs, une planification minutieuse, une opérationnalisation précise et un entretien constant. Il faut aussi noter que les divers standards associés aux *Web Services* et à XML sont encore en mouvance. Cela peut donc créer des difficultés au niveau de l'interopérabilité promise par les *Web Services*.

Dans ce qui suit, nous discutons des limites les plus marquantes des *Web Services*.

5.1 Sémantique des messages

Les *Web Services* fournissent une solution pour le transport de messages entre deux partenaires. Cependant, ils ne décrivent pas comment interpréter les messages transmis. Il appartient donc aux partenaires de s'entendre sur cette sémantique. Pour illustrer ce problème, prenons l'exemple d'un individu qui désire effectuer un voyage. Pour ce faire, il voudra acheter tous les services nécessaires à ce périple sur la page Web de son agence de voyage. Chaque service est fourni par un fournisseur distinct, qui désigne l'individu différemment. Ainsi :

- le transporteur désigne l'individu comme un « passager »,
- l'hôtelier désigne l'individu comme un « invité »,
- le marchand de valise désigne l'individu comme un « client »,
- l'assureur désigne l'individu comme un « assuré » et
- l'hôpital (fournissant un vaccin) désigne l'individu comme un « patient ».

Or, il s'agit toujours du même individu. L'agence de voyage devant transiger avec chacun des fournisseurs au nom de son « client » (l'individu), il est clair qu'elle devra faire le lien entre les différentes désignations. Évidemment, nous n'avons considéré ici que les transactions unilingues. Cela se complique lorsque plusieurs langues et cultures sont impliquées.

« ... comme tout effort technologique, l'implantation des Web Services requièrent des objectifs clairs, une planification minutieuse, une opérationnalisation précise et un entretien constant. »

⁷⁷ Traduction libre de : O'Reilly Network: *Web Services - An Executive Summary* [Apr. 12, 2002], <http://www.oreillynet.com/pub/a/webservices/2002/04/12/excreport.html>.

À la lumière de cet exemple, il est clair que des mécanismes de description sémantique sont nécessaires afin de réconcilier ces différences de désignation. Déjà plusieurs solutions sont envisagées afin de résoudre ces problèmes. Par exemple, ebXML introduit la notion de *core components*⁷⁸, qui permet d'identifier de façon unique un concept, même si ce concept peut prendre différentes appellations dans différents contextes. Ainsi, la notion de client pourra facilement être associée aux termes passager, invité, client, assuré, etc. Cette idée de fixer la sémantique des concepts est aussi à l'origine de plusieurs dictionnaires spécialisés, utilisés pour les échanges XML dans plusieurs domaines. C'est aussi l'approche favorisée par RosettaNet.

Les efforts de recherche sur le Web sémantique entrent aussi dans la liste des solutions envisageables. On notera RDF⁷⁹ (*Resource Description Framework*) et DAML+OIL⁸⁰ (*DARPA Agent Markup Language + Ontology Inference Layer*) qui permettent la description de contenu sémantique. Ces descriptions devraient permettre d'effectuer la réconciliation sémantique nécessaire au bon fonctionnement des Web Services.

5.2 Processus d'affaire inter-organisationnel

Dans leur état actuel, les *Web Services* fournissent une infrastructure logicielle pour les interactions inter-organisationnelles. Cependant, ils ne permettent pas de définir des processus d'affaire inter-organisationnels. Pour ce faire, il faut utiliser des langages (p.ex., WSFL⁸¹, WSCL⁸², Xlang⁸³) et des architectures se trouvant à des niveaux d'abstraction plus élevés. Parmi ces architectures, on retrouve notamment :

- l'architecture RosettaNet qui décrit un grand nombre de processus d'affaires, spécifiant comment les partenaires peuvent interagir,
- l'architecture ebXML qui permet la définition de processus inter-organisationnel, la description des compétences d'une entreprise et l'établissement d'ententes de collaboration.

Au chapitre de la standardisation de ces types de langages et d'architecture la grande difficulté sera de s'entendre sur un processus qui répond à la fois aux besoins des grandes entreprises et à ceux des PME.

« Dans leur état actuel, les Web Services fournissent une infrastructure logicielle pour les interactions inter-organisationnelles. Cependant, ils ne permettent pas de définir des processus d'affaire inter-organisationnels. »

« Au chapitre de la standardisation de ces types de langages et d'architecture la grande difficulté sera de s'entendre sur un processus qui répond à la fois aux besoins des grandes entreprises et à ceux des PME. »

⁷⁸ Kotok et. al., ebXML: The new global standard for doing business over the internet, éd. New Riders, sept. 2001, ISBN 0-7357-1117-8.

⁷⁹ <http://www.w3.org/RDF/>

⁸⁰ <http://www.daml.org/>

⁸¹ Cover Pages: *Web Services Flow Language (WSFL)*, <http://xml.coverpages.org/wsfl.html>.

⁸² *Web Services Conversation Language (WSCL) 1.0*, <http://www.w3.org/TR/wscl10/>.

⁸³ XLANG, http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm.

5.3 Sécurité

Un défi important des *Web Services* est leur sécurisation. Présentement, les *Web Services* dépendent uniquement de la sécurité offerte par les mécanismes de transport utilisés, tels SSL (*Secure Socket Layer*), l'utilisation d'un réseau virtuel privé (VPN) ou HTTPS. Ces mécanismes n'offrent habituellement qu'un service de confidentialité des données. Le commerce électronique requiert l'intégrité des données, la non-répudiation et le contrôle d'accès⁸⁴ ce qui n'est pas nécessairement garanti par les mécanismes de transport.

Plusieurs technologies et protocoles sont déjà à l'étude et/ou utilisées afin de sécuriser les *Web Services*. Parmi ces technologies, on notera SAML⁸⁵ (*Security Assertion Markup Language*) développé par OASIS et WS-Security⁸⁶ (*Web Services-Security*) développés par IBM, Microsoft et Verisign.

5.4 Aspects légaux

La venue des *Web Services* introduit de nombreuses questions de nature légale⁸⁷ :

- Qu'elle est la nature de la relation entre les entreprises utilisant un *Web Services*? Quel est le niveau de cohérence et de couplement (léger ou fort)?
- Quel est le niveau de confiance qu'ils partagent?
- Qu'elles sont les nouvelles frontières entre entreprises dans un contexte *Web Services*?
- À qui appartient quoi dans un contexte d'échange *Web Services*?
- Qu'elles sont les attentes légales de chaque entreprise transigeant à l'aide des *Web Services*?
- Quels sont les risques associés à la responsabilité de chacune des parties dans une transaction *Web Services*? À quel moment précis d'un processus cette responsabilité prend-elle effet? Y a-t-il un partage des risques? Qui est ultimement imputable de la responsabilité?
- Comment assurer la qualité de service par un processus commun?

« La venue des Web Services introduit de nombreuses questions de nature légale. »

⁸⁴ V. Hassler, *Security Fundamentals for E-commerce*. Computer Security Series, Artech House, 2001.

⁸⁵ Cover Pages: Security Assertion Markup Language (SAML), <http://xml.coverpages.org/saml.html>.

⁸⁶ *Web Services Security* (WS-Security), <http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>.

⁸⁷ Présentation de Matthew K. Hettinger, *Legal Considerations for Web Services, Federated Systems and E-Commerce*, Copyright 2002 Mathet Consulting, Inc. 1450 E. American Lane, PMB 14004, Schaumburg, IL 60173.

- Jusqu'à quel point un contrat légal peut-il être modelé et intégré dans un système fédéré?

Ces questions et bien d'autres viendront garnir la réflexion juridique de chacune des entreprises transigeant avec d'autres entreprises à l'aide des *Web Services*. Ces questions devront éventuellement trouver réponses au sein des organismes de standardisation, des forums internationaux et dans les entreprises.

6 Conclusion

Les *Web Services* sont le résultat de la collaboration exceptionnelle des joueurs majeurs des technologies de l'information qui se sont entendus sur un certain nombre de protocoles et d'approches qui favoriseront l'interopérabilité entre les plateformes, les systèmes d'exploitation, les langages et les programmes. De ce fait, les *Web Services* sont donc plus un phénomène ou un concept, voire un contexte, qu'une technologie. Il est évident que les *Web Services* reposent sur diverses technologies, mais ils représentent surtout une volonté commune des manufacturiers, des organismes de standards et des utilisateurs de développer des outils permettant une réelle interopérabilité.

Fonctionnellement, les *Web Services* n'apportent rien de neuf par rapport aux *middlewares*, conçus pour appeler des services distants, ou par rapport à l'EDI, qui décrit comment échanger des documents d'affaires. Cependant, la façon avec laquelle les *Web Services* réalisent ces fonctions est entièrement novatrice, découplant le service lui-même de son implantation. Ceci nous porte à croire que les *Web Services* devraient faciliter l'émergence de processus d'affaires automatisés, souples, dynamiques et répondant aux besoins spécifiques des clients et partenaires et des entreprises. De plus, dans la mesure où les efforts de recherche permettant de décrire les aspects sémantiques des échanges XML (*core components* de ebXML, RDF, DAML+OIL, etc.) portent fruits, l'utilisation des *Web Services* sera simplifiée.

À notre avis, les *Web Services* sont suffisamment développés pour que les entreprises les utilisent dès maintenant, afin de récolter les divers bénéfices de la technologie. L'utilisation rapide de la technologie permettra à ces entreprises de développer à l'interne une compréhension et une expertise, qui pourra alors être utilisée à l'externe une fois que tous les standards seront déterminés. Par le biais des *Web Services*, ces entreprises pourront assouplir les liens entre leurs applications et celles de leurs partenaires, créant des systèmes légèrement couplés et plus flexibles. Cet assouplissement doit passer par une modélisation en profondeur des actifs logiciels de l'entreprise afin de bien identifier les services et ressources qu'elle doit déployer. En procédant de cette façon, l'entreprise optimise les avantages des *Web Services*; elle identifiera les applications, informations et processus d'affaire à valeur ajoutée qu'elle aura avantage à rendre disponible à ses partenaires.

« Il est évident que les Web Services reposent sur diverses technologies, mais ils représentent surtout une volonté commune des manufacturiers, des organismes de standards et des utilisateurs de développer des outils permettant une réelle interopérabilité. »

À plus long terme, les *Web Services* deviendront plus qu'une simple technologie d'intégration faiblement couplée. Ils deviendront la pierre angulaire des interactions inter-organisationnelles. Pour ce faire, les services déployés à l'aide des *Web Services* devront s'insérer dans un cadre d'utilisation plus global, tel que prescrit par ebXML. Ainsi une solution idéale combinerait l'utilisation de l'architecture ebXML pour définir le contexte des échanges et la sémantique des message, d'une part, et l'utilisation des *Web Services* comme mécanisme légèrement couplé pour la transmission des messages et l'invocation des services.

« Ainsi une solution idéale combinerait l'utilisation de l'architecture ebXML pour définir le contexte des échanges et la sémantique des message, d'une part, et l'utilisation des Web Services comme mécanisme légèrement couplé pour la transmission des messages et l'invocation des services. »

7 Annexe 1 Lexique⁸⁸

- **API** (*Application Program Interface*) : Méthode spécifique d'un système d'opération ou d'une application particulière, permettant qu'un programmeur puisse faire une requête à ce système d'opération ou application particulière à partir d'une autre application qu'il est en train de coder.
- **Basic** : Un des premiers langages de programmation. Il est toujours utilisé de nos jours.
- **Balises** (*tag*) : Terme générique identifiant les éléments descriptifs d'une chaîne de codes informatiques. Par exemple, dans la suite de code: <title> Les Web Services et leur impact sur le commerce B2B</title>, les pictogrammes et les mots *title* sont les balises.
- **BPSS** (*Business Process Specification Schema*) : Architecture standardisée par laquelle une application d'affaires peut être configurée pour permettre la conclusion d'une transaction d'affaires informatisée. Cette architecture est l'une des sous-composantes de ebXML et a été développée sous l'égide de UN/CEFACT.
- **B2B** (*Business-to-Business*) : Qualificatif signifiant une interaction entre deux entités d'affaires. Par exemple, nous pouvons utiliser l'expression marketing B2B, qui signifie le marketing d'entreprise à entreprise.
- **B2BI** (*Business-to-Business Integration*) : Terme se référant à des logiciels, des entreprises ou des applications permettant l'intégration d'applications d'affaires entre des entreprises.
- **C** : Langage de programmation structuré et procédural qui est largement utilisé afin de programmer des systèmes d'opérations ainsi que des applications.

⁸⁸ Les définitions de ce lexique sont tirées et adaptées du site Whatis.com. <http://whatis.techtarget.com>.

- **C++** : Langage de programmation orienté objet. Ce langage de programmation est une version plus évoluée et ayant pris naissance dans le langage C.
- **C#** (*C sharp*) : Langage de programmation orienté objet. Ce langage de programmation est développé par Microsoft et est une version découlant de C++ et de Visual Basic (de Microsoft).
- **CORBA** (*Common Object Request Broker Architecture*) : Architecture et spécifications visant à permettre de créer, publier et gérer des objets, sur un réseau, dans un contexte de programmation distribuée. Cette architecture a déjà été largement utilisée sauf par Microsoft qui avait développé sa propre architecture DCOM.
- **Coupe-feu** : Ensemble de logiciels et matériel localisé à l'entrée d'un réseau et visant à protéger les ressources d'un réseau des utilisateurs d'autres réseaux.
- **Couplage** : Le couplage en informatique est associé à l'idée de préconnaissance des spécifications matérielles et logicielles d'applications d'affaires à intégrer ensemble. Ainsi, s'il faut connaître ces spécificités avant de pouvoir programmer la communication entre deux applications, elles seront fortement couplées tandis que dans le cas des *Web Services* où il n'est pas nécessaire de connaître ses spécificité, elles seront donc légèrement couplées.
- **CPP/A** (*Collaboration Protocol Profile / Agreement*) : Un des sous éléments de l'architecture ebXML. Il vise à permettre l'interopérabilité universelle entre deux parties voulant conclure une entente d'affaires sur support électronique. Le CPP décrit les capacités techniques et les types de collaboration que le message supporte, tandis que le CPA définit l'entente qui sera conclue entre les deux parties.
- **CTO** (*Chief Technology Officer*) : Chef ou vice-président des technologies de l'information au sein d'une entreprise.
- **DAML+OIL** (*DARPA Agent Markup Language + Ontology Inference Layer*) : Langage permettant de décrire des concepts, leurs caractéristiques ainsi que les relations sémantiques entre concepts. C'est un langage utilisant la syntaxe XML.
- **DCOM** (*Distributed Component Object Model*) : Ensemble d'interfaces programmes et de concepts développé par Microsoft qui est en fait l'équivalent de CORBA. Il est utilisé entre autre pour faire des appels RPC (Remote Procedure Call).
- **ebXML** (*Electronic Business XML*) : Architecture de commerce électronique visant à utiliser XML pour permettre les échanges de données entre les entreprises. C'est une tentative de standardisation d'une architecture développée par OASIS et UN/CEFACT et dont l'un des objectifs est de construire sur les paramètres existant d'un autre standard soit celui de l'EDI.

- **EAI** (*Enterprise Application Integration*) : Terme se référant à des logiciels, des entreprises ou des applications permettant l'intégration d'applications d'affaires à l'intérieur d'une entreprise. Par exemple les EAI servent à intégrer un SCM avec un CRM.
- **EDI** (*Electronic Data Interchange*) : Standard décrivant les formats utilisés dans l'échange de données d'affaires entre entreprises.
- **ERP** (*Entreprise Resource Planning*) : Ensemble d'applications d'affaires permettant la gestion intégrée des ressources de l'entreprise. Un système ERP permet en effet de gérer dans une même interface utilisateur, les achats, l'inventaire, les ressources humaines, les listes fournisseurs et le suivi des commandes.
- **Granularité** : Grosseur, échelle, niveau de détail ou profondeur de pénétration relative d'un objet ou d'une activité informatique.
- **HTML** (*Hypertext Markup Language*) : Ensemble de balises et de symboles insérés dans un fichier informatique visant à être affiché sur un navigateur World Wide Web. Les balises HTML définissent comment un navigateur Web devra afficher les informations contenues dans une page Web. HTML a été développé par le W3C.
- **HTTP** (*Hypertext Transfer Protocol*) : Protocole de communication permettant l'échange de fichiers sur le World Wide Web. HTTP est un protocole d'application développé par le W3C.
- **HTTPS** (*Hypertext Transfer Protocol over Secure Socket Layer, or HTTP over SSL*) : Protocole HTTP auquel l'entreprise Netscape a ajouté la couche SSL pour en augmenter la sécurité. HTTPS est un protocole permettant l'échange sécurisé de fichiers sur le World Wide Web.
- **IETF** (*Internet Engineering Task Force*) : Organisme de standardisation supervisé par l'Internet Society et son Internet Architecture Board (IAB). L'IETF définit des protocoles ayant trait aux opérations sur Internet tel que les protocoles TCP/IP ou SMTP.
- **Interfaces** : Une interface peut décrire trois différents concepts, selon le contexte :
 - une interface utilisateur, c'est-à-dire un ensemble de boutons, de graphiques, de cadrans ou de commandes qu'un programme ou un ordinateur présente à un utilisateur pour lui permettre d'interagir et d'utiliser le programme ou l'ordinateur;
 - une interface de programmation, c'est-à-dire un ensemble de fonctions, de balises, d'options et de codes informatique permettant à un programmeur d'interagir avec le programme;
 - le lien logique ou physique reliant deux composantes matérielles l'une à l'autre ou à un branchement.

- **Java** : Langage de programmation orienté objet développé expressément pour être utilisé dans un contexte d'environnement distribué. Ce langage a été développé par l'entreprise Sun Microsystems.
- **Liaisons** (*binding*) : En programmation informatique, action de relier ensemble deux ou plusieurs objets d'un programme, ou d'une valeur d'un document durant une période définie de temps et d'espace.
- **OASIS** (*Organization for the Advancement of Structured Information Standards*): OSBL technologique international visant la promotion et l'adoption de standards indépendants des manufacturiers, dans le domaine des technologies de l'information. OASIS tente de rapprocher les acteurs industriels et les groupes de standardisation afin qu'ils s'entendent sur l'utilisation universelle de XML.
- **PDA** (*personal digital assistant*): Terme générique qui représente l'ordinateur de poche.
- **Programmation orienté objet** (POO): Concept qui a révolutionné les règles de développement logiciel. Traditionnellement, un programme est un processus logique permettant de transformer un input en un output selon une séquence particulière. La programmation orientée objet, quant à elle, s'intéresse aux données elle-même qui sont perçues comme des objets que l'on manipule plutôt que sur le processus de manipulation lui-même.
- **Protocole** : Ensemble de règles qui sont partagée entre deux ordinateurs afin de communiquer l'un avec l'autre.
- **RDF** (*Resource Description Framework*) : Langage permettant de décrire le contenu sémantique d'un document HTML.
- **Registre** : Endroit unique servant à entreposer des informations nominales ayant trait à des caractéristiques physiques et logicielles (tel que dans un système d'opération Window 95 et +), et des informations sur une entreprise (tel que dans les registres de ebXML ou UDDI).
- **RMI** (*Remote Method Invocation*) : Manière qu'utilise un programmeur dans un environnement Java pour écrire une programmation orientée objet et permettre aux objets qui sont sur différents serveurs d'interagir dans un environnement distribué. RMI est la version Java de RPC.
- **RosettaNet** : Organisation qui regroupe plusieurs des acteurs majeurs de l'industrie de TI. La contribution majeure de cette organisation se situe au niveau de la recherche et de l'établissement de standards commun de processus d'affaires et d'un dictionnaire d'affaires électroniques universel nommément appelé infrastructure RosettaNet.
- **RPC** (*Remote Procedure Call*) : Protocole permettant à un programme de requérir un service d'un autre programme situé dans le serveur d'un autre réseau, sans avoir à connaître tout les détails techniques de ce réseau au préalable. Il s'agit d'un protocole utilisé dans une architecture client/serveur.

- **SAML**⁸⁹ (*Security Assertion Markup Language*) : Standard de la famille XML qui permet à un utilisateur de ne s'authentifier qu'une seule fois pour avoir accès à des fédérations de sites Web qui sont pourtant séparés l'un de l'autre. OASIS (*Organization for the Advancement of Structured Information Standards*) : est l'organisme de standardisation qui chapeaute le standard qui est supporté dans l'architecture ebXML et l'environnement Biztalk de Microsoft.
- **SCM** (*Supply Chain Management*) : Logiciel ou série de logiciels d'applications d'affaires visant à permettre la gestion continue des inventaires. Une application SCM permet généralement de gérer les flux de produits, d'informations et financiers d'une entreprise.
- **SGML** (*Standard Generalized Markup Language*) : Ancêtre de XML. SGML sert à décrire le langage d'un document. Il n'est pas un langage de document en lui-même. C'est plutôt une série de métadonnées indiquant comment un langage doit spécifier les tags qui le composent.
- **SMTP** (*Simple Mail Transfer Protocol*) : Protocole de transfert et de réception de courriel largement utilisé dans le monde. L'IETF est l'organisme de standardisation qui chapeaute ce protocole.
- **SOI** (*Service Oriented Integration*) : Architecture orientée services, développée par l'industrie des TI pour répondre aux problèmes d'intégration des applications d'affaires. L'idée est d'envelopper une application d'affaires dans une interface définie et de présenter cette interface sur un réseau. L'application devient ainsi un service que l'on peut jumeler à d'autres services. Ce jumelage de plusieurs services (d'applications d'affaires ayant une interface commune) pour en faire un bus continue d'information est ce que l'on nomme l'architecture orientée service.
- **SOAP** (*Simple Object Access Protocol*) : Protocole de base des *Web Services*. Il sert à l'échange d'informations dans un environnement distribué et décentralisé. Il permet à un programme résidant dans un système d'opération de communiquer avec un autre programme résident d'un système d'opération complètement différent.
- **SI** (Système d'information) : Ensemble des ressources technologiques et humaines contribuant à l'entreposage, le traitement de données, la distribution et la communication des informations requises par une partie ou la totalité d'une entreprise.
- **SSL** (*Secure Socket Layer*) : Protocole permettant l'établissement d'un canal de communication sécurisée sur Internet. Bien que le protocole ait été développé par Netscape, il jouit néanmoins d'un large consensus de l'industrie informatique incluant même l'appui de Microsoft.

⁸⁹ Cover Pages: Security Assertion Markup Language (SAML), <http://xml.coverpages.org/saml.html>.

- **UBR** (*Universal Business Registry*) : Registre résident sur les serveurs d'une entreprise mais disponible au monde entier pour recevoir les informations UDDI se rattachant à un *Web Service*.
- **UDDI** (*Universal Description, Discovery and Integration*) : Spécification XML chapeautée par l'organisme OASIS régissant l'information relative à la publication, la découverte et l'utilisation d'un *Web Service*.
- **UN/CEFACT** (*United Nations Centre for Trade Facilitation and Electronic Business*) : Organisme supranational des Nations Unies visant à faciliter et à accroître le commerce au niveau mondial. Son mandat est d'être un facilitateur mondial des transactions internationales en participant à l'harmonisation et à la simplification des processus et flux d'informations. UN/CEFACT a été l'un des promoteurs de l'architecture ebXML.
- **URI** (*Uniform Resource Identifier*) : identifiant d'un point de contenus sur le Web . Le URI le plus connu est le URL (*Uniform Resource Locator*) qui est en fait l'adresse d'une page Web. Un URI décrit typiquement le mécanisme utilisé pour accéder à une ressource sur le Web, l'ordinateur spécifique où cette ressource est logée et le nom de fichier spécifique sur cet ordinateur.
- **VAN** (*Value Added Network*) : Réseau privé utilisé par une entreprise pour établir un lien de données dédié et servant de lien pour le transfert de données de type EDI ou d'autres services réseaux. Avec la venue du World Wide Web, les entreprises ont de moins en moins recours au VAN pour l'acheminement des données.
- **Visual Basic** : Langage et environnement de programmation de Microsoft incluant une interface graphique pour l'utilisateur, lui permettant de choisir et de modifier des sections de codes informatique écrits dans le langage Basic.
- **VPN** (*virtual private network*) : Manière d'utiliser une infrastructure publique de communication (comme Internet) pour fournir un lien privé et sécurisé à des utilisateurs distants.
- **WSCL** (*Web Services Conversation Language*) : Une des nombreuses tentatives de standardisation d'un langage du flux des processus d'affaires devant compléter éventuellement le corpus des standards *Web Services* déjà établi. Ce langage est chapeauté par le W3C.
- **WSDL** (*Web Services Description Language*) : Un des standards du corpus *Web Services*. Il sert à décrire et publier le format et les protocoles d'un *Web Service* de manière homogène par l'utilisation du format XML. Il est chapeauté par W3C.
- **WSFL** (*Web Services Flow Language*) : Une des nombreuses tentatives de standardisation d'un langage du flux des processus d'affaires devant compléter éventuellement le corpus des standards *Web Services* déjà établi. Ce langage est chapeauté par OASIS.

- **WS-I Organization** (*Web Services Interoperability*) : Association d'entreprises de l'industrie des TI visant à créer des spécifications *Web Services* que tous les fabricants peuvent utiliser. Cet organisation fondée par IBM et Microsoft est la réponse des deux géants face à la lenteur qu'ils perçoivent du W3C.
- **WS-Security** (*Web Services-Security*) : Une des tentatives de standardisation d'un mécanisme de sécurisation des données échangées à l'aide des *Web Services*. Cette spécification est proposée par IBM, Microsoft et Verisign et est l'une des activités de WS-I Organization.
- **W3C** (*World Wide Web Consortium*) : Consortium industriel visant à favoriser l'interopérabilité des produits informatique et l'évolution du World Wide Web, par le développement de standards et de spécifications techniques.
- **Xlang** (*XML business process language*) : Une des nombreuses tentatives de standardisation d'un langage du flux des processus d'affaires devant compléter éventuellement le corpus des standards *Web Services* déjà établi. Ce langage est développer par Microsoft.
- **XML** (*eXtensible Markup Language*) : Collection de standards comptant plus de 450 spécifications individuelles et servant à faciliter le partage des formats de données et des données sur le World Wide Web.

8 Annexe 2

8.1 Lectures complémentaires suggérées

- Hagel III, Brown, article: Break on Through to the Other Side: A Missing Link in Redefining the Enterprise, 2002, site Web www.johnhagel.com.
- Kotok et. al., ebXML: The new global standard for doing business over the internet, éd. New Riders, sept. 2001, ISBN 0-7357-1117-8.
- Newcomer, Eric, Understanding Web Services: XML, WSDL, SOAP and UDDI, éd. Addison-Wesley, 2002.
- Oellermann, Jr., William, Architecting Web Services, éd. Apress, 2002.

8.2 Bibliographie additionnelle

- Batchelder, R, The SMB Internet Scenario, Gartner Research, Research Note may 2001, COM-13-4497.

- Bloomberg et.al., article Sun Microsystems: Left behind at *Web Services* Altar?, ZapThink opinion, avril 2002.
- Deloitte & Touche, The Blue Paper, *Web Services: The next evolution in software*, 2002.
- Dyck, Timothy, article: *Web Services* impact, eWeek, Ziff Davis Media, Vol. 19, numéro 37.
- Hagel III, Brown, Break on Through to the Other Side: A Missing Link in Redefining the Enterprise, 2002.
- Hagel III et. al., Orchestrating Loosely Coupled Business Processes: The Secret to Successful Collaboration, 2002.
- Hagel III, John, Out of the box: Strategies for achieving profits today and growth tomorrow through *Web Services*, Harvard Business school press, 2002.
- Kotok et. al., ebXML: The new global standard for doing business over the internet, éd. New Riders, sept. 2001, ISBN 0-7357-1117-8.
- Newcomer, Eric, Understanding *Web Services* : XML, WSDL, SOAP and UDDI, éd. Addison-Wesley, 2002.
- Oellermann, Jr., William, Architecting *Web Services*, éd. Apress, 2001.
- Patricia Seybold Group, An executive's guide do *Web Services*, How to optimize *Web Services* investments to improve your customer experience, Patricia Seybold Group's executive series, 2002.
- Plummer, Daryl, article: Key entry points into *Web Services* markets, Gartner, Article Top View, févr. 2002, no. AV-15-5191.
- Schmelzer et.al. article : Understanding the Real Costs of Integration, ZapFlash, Zapthink Research, oct. 2003.
- SeeBeyond, *Web Services*, A SeeBeyond White Paper, 2002.
- Slywotsky, Adrian J., et. al., How digital is your business? éd. Crown Business, 2000.
- Smith, David, Gartner research, Explaining *Web Services* Apparent Contradictions, Article Top View, AV-16-4551, juin 2002.
- Smith, David, Software Vendors Weave *Web Services* Into Their .zStrategies, Gartner Research, nov. 2001, AV-14-8859.
- The Aberdeen Group, The next B-toB Gestalt: Business Process Networks, Market Viewpoint, Vol. 15, no.2, March 2002.
- The Stencil Group, The Evolution of UDDI UDDI.org White Paper, The Stencil Group, juillet 2002.
- The Stencil Group, Why UDDI Will Succeed, Quietly: Two Factors Push *Web Services* Forward, The Stencil Group, avril 2001.
- Yellin, Daniel M., Stuck in the Middle: Challenges and Trends in Optimizing Middleware, IBM T. J. Watson Research Center, Hawthorne, NY 10532 .
- ZapThink, LLC, article : The "Pros and Cons" of XML, coll. ZapThink research report, 2001.
- ZapThink, LLC, Poster Key XML specifications, Zapthink Document IDZTS-G1101, mai 2002.

8.3 Webographie technologique

Pour une vision technologique d'XML vous pouvez consulter les sites :

- Extensible Markup Language (XML), <http://www.w3.org/XML/>
- Extensible Markup Language (XML), <http://www.w3.org/TR/REC-xml>
- Namespaces in XML, <http://www.w3.org/TR/1999/REC-xml-names-19990114/>
- XML Linking Language (XLink), <http://www.w3.org/TR/xlink/>
- XML Pointer Language (XPointer), <http://www.w3.org/TR/xptr/>
- XML Path Language (XPath), <http://www.w3.org/TR/xpath>
- *XML Schema*
 - Part 0: Primer, <http://www.w3.org/TR/xmlschema-0/>
 - Part 1: Structures, <http://www.w3.org/TR/xmlschema-1/>
 - Part 2: Datatypes, <http://www.w3.org/TR/xmlschema-2/>
- XQuery: A Query Language for XML, <http://www.w3.org/TR/xquery/>
- XML Protocol Comparisons, <http://www.w3.org/2000/03/29-XML-protocol-matrix>
- XML en 10 points, <http://www.w3.org/XML/1999/XML-in-10-points>

Pour une vision technologique de SOAP vous pouvez consulter les sites :

- Simple Object Access Protocol (SOAP) 1.1, <http://www.w3.org/TR/SOAP/>
- *SOAP Version 1.2*
 - Part 0: Primer, <http://www.w3.org/TR/soap12-part0/>
 - Part 1: Messaging Framework, <http://www.w3.org/TR/soap12-part1/>
 - Part 2: Adjuncts, <http://www.w3.org/TR/2001/WD-soap12-part2-20011217/>
- SOAP 1.2 Attachment Feature, <http://www.w3.org/TR/2002/WD-soap12-af-20020814/>
- SOAP Version 1.2 Email Binding, <http://www.w3.org/TR/2002/NOTE-soap12-email-20020626>
- SOAP Binding to Email (RFC2822 - Internet Message Format), <http://www.w3.org/2000/xp/Group/2/02/emailbinding.html>
- The "application/soap+xml" media type, <http://www.w3.org/2000/xp/Group/2/06/18/draft-baker-soap-media-reg-01.txt>
- SOAP Version 1.2 Specification Assertions and Test Collection, <http://www.w3.org/TR/soap12-testcollection.html>
- SOAP Messages with Attachments, <http://www.w3.org/TR/SOAP-attachments>
- MIME (Multipurpose Internet Mail Extensions), <http://www.nacs.uci.edu/indiv/ehood/MIME/MIME.html>

- SoapWare.Org : Implementations, <http://www.soapware.org/directory/4/implementations>
- Anatomy of a SOAP Call, <http://www.devx.com/upload/free/features/entdev/1999/11nov99/cv1199/cv1199.asp>

Pour une vision technologique de WSDL vous pouvez consulter les sites :

- Version 1.1 Web Service Definition Language (WSDL), <http://www.w3.org/TR/wsdl>
- Version 1.2 Web Services Description Language (WSDL) Version 1.2, <http://www.w3.org/TR/wsdl12/>
- Web Services Description Working Group, <http://www.w3.org/2002/ws/desc/>
- WSDL Tutorial, <http://www.w3schools.com/wsdl/default.asp>
- WSDL Specification Index Page, Microsoft, Welcome to the MSDN Library, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwsdl/html/wsdlspecindex.asp>
- Cover Pages: Web Services Description Language (WSDL), <http://xml.coverpages.org/wsdl.html>
- Web Services Description Language (WSDL) 1.0, <http://xml.coverpages.org/wsdl20000929.html>
- Introduction to WSDL, <http://www.learnxmlws.com/tutors/wsdl/wsdl.aspx>
- O'Reilly Network: Emerging Technology Briefs: WSDL [May. 01, 2002], <http://www.oreillynet.com/pub/a/webservices/2002/04/30/wsdl.html>

Pour une vision technologique de UDDI vous pouvez consulter les sites :

- UDDI.org, <http://www.uddi.org/>
- UDDI Version 2.0 XML Schema, http://www.uddi.org/schema/uddi_v2.xsd
- UDDI Version 2.0 XML Replication Schema, http://www.uddi.org/schema/uddi_v2replication.xsd
- UDDI Version 2.0 XML Custody Schema, http://www.uddi.org/schema/uddi_v2custody.xsd
- UDDI Version 3, <http://www.uddi.org/specification.html>
- UDDI V3 Specification, http://uddi.org/pubs/uddi_v3.htm#_Toc12653608
- Web Services Body, UDDI.org, Transitions Work to OASIS Standards Consortium UDDI.org, http://www.uddi.org/news/uddi_news_07_30_02.html
- OASIS - News - 08_28_2002, http://www.oasis-open.org/news/oasis_news_08_28_02.shtml
- The Evolution of UDDI UDDI.org White Paper, The Stencil Group, juillet 2002, http://www.uddi.org/pubs/the_evolution_of_uddi_20020719.pdf
- Registres publics: UDDI.org, <http://www.uddi.org/register.html> et UDDI-China.ORG - Universal Description, Discovery and Integration, <http://www.uddi-china.org/register/>

Pour une vision technologique de ebXML vous pouvez consulter les sites :

- *ebXML Technical Architecture Specification v1.04*, <http://www.ebxml.org/specs/ebTA.pdf>
- *Technical Architecture Risk Assessment v1.0*, <http://www.ebxml.org/specs/secRISK.pdf>
- *Proposed revisions to Technical Architecture Specification v1.0.4*, <http://www.ebxml.org/specs/bpTAREV.pdf>
- *Collaboration-Protocol Profile and Agreement Specification v1.0*, <http://www.ebxml.org/specs/ebCCP.doc>
- *Business Process Specification Schema v1.01*, <http://www.ebxml.org/specs/ebBPSS.pdf>
- *Business Process and Business Information Analysis Overview v1.0*, <http://www.ebxml.org/specs/bpOVER.pdf>
- *Business Process Analysis Worksheets & Guidelines v1.0*, <http://www.ebxml.org/specs/bpWS.pdf>
- *E-Commerce Patterns v1.0*, <http://www.ebxml.org/specs/bpPATT.pdf>
- *Catalog of Common Business Processes v1.0*, <http://www.ebxml.org/specs/bpPROC.pdf>
- *Core Component Overview v1.05*, <http://www.ebxml.org/specs/ccOVER.pdf>
- *Core Component Discovery and Analysis v1.04*, <http://www.ebxml.org/specs/ebCCDA.doc>
- *Naming Convention for Core Components v1.04*, <http://www.ebxml.org/specs/ebCCNAM.pdf>
- *Guide to the Core Components Dictionary v1.04*, <http://www.ebxml.org/specs/ccCTLG.pdf>
- *Core Component Dictionary v1.04*, <http://www.ebxml.org/specs/ccDICT.pdf>
- *Core Component Structure v1.04*, <http://www.ebxml.org/specs/ccSTRUCT.pdf>
- *Context and Re-Usability of Core Components v1.04*, <http://www.ebxml.org/specs/ebCNTXT.pdf>
- *Document Assembly and Context Rules v1.04*, <http://www.ebxml.org/specs/ebCCDOC.pdf>
- *Catalogue of Context Drivers v1.04*, <http://www.ebxml.org/specs/ccDRIV.pdf>
- *Registry Information Model v1.0*, <http://www.ebxml.org/specs/ebRIM.pdf>
- *Registry Services Specification v1.0*, <http://www.ebxml.org/specs/ebRS.pdf>
- *Registry Services Specification v2*, <http://www.ebxml.org/specs/ebRS2.pdf>
- *Using UDDI to find ebXML*, <http://www.ebxml.org/specs/rrUDDI.doc>
- *Registry/Repository*, <http://www.ebxml.org/specs/rrUDDI.doc>
- *ebXML Registry Security Proposal*, <http://www.ebxml.org/specs/secREG.pdf>
- *Message Service Specification v1.0*, <http://www.ebxml.org/specs/ebMS.pdf>
- *Message Service Specification v2*, <http://www.ebxml.org/specs/ebMS2.pdf>
- *ebXML Glossary*, <http://www.ebxml.org/specs/ebGLOSS.pdf>
- *ebXML Technical Architecture Specification v1.04*, <http://www.ebxml.org/specs/ebTA.pdf>

Pour une vision technologique de WSFL vous pouvez consulter le site :

- <http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>

Pour une vision technologique de WSCL vous pouvez consulter le site :

- Web Services Conversation Language (WSCL) 1.0,
<http://www.w3.org/TR/wscl10/>

Pour une vision technologique de Xlang vous pouvez consulter le site :

XLANG, http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm