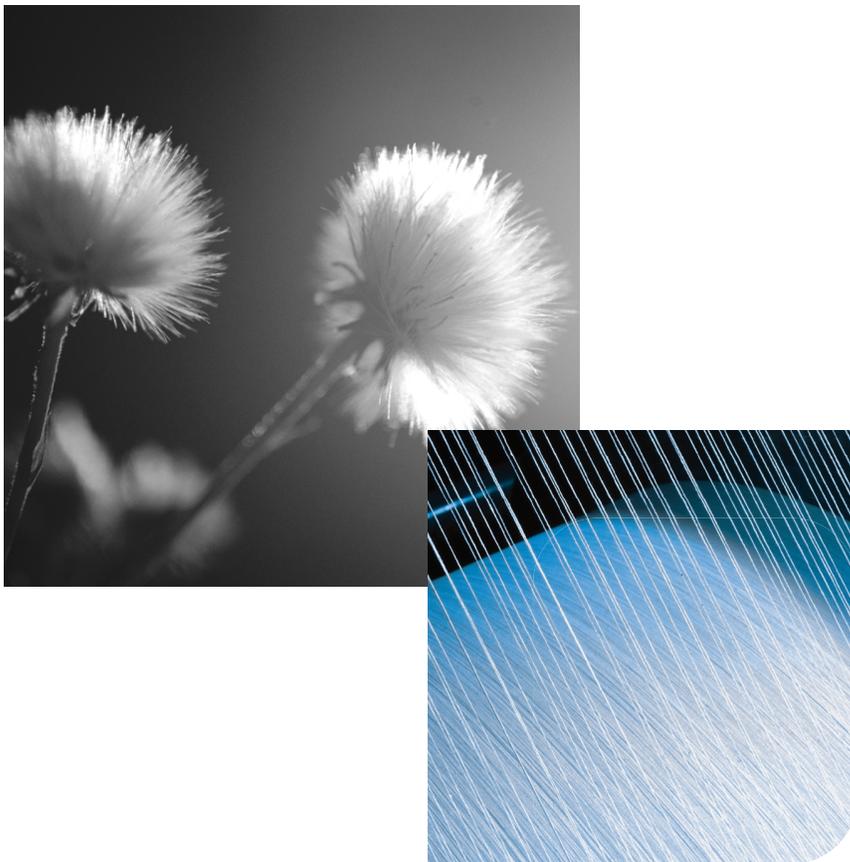




SOA et urbanisme

**Le rôle des Architectures Orientées Services dans
l'alignement métier des Systèmes d'Information**



Auteur et contributeurs

François Rivard – manager de la Business Unit e-Integration d'Unilog Management, François Rivard assiste les entreprises dans la qualification de leur besoin SOA/EAI, dans le cadrage de la démarche à adopter et dans le choix des produits. Il est l'auteur de trois ouvrages consacrés aux nouvelles technologies, à l'EAI et à ses enjeux pour les systèmes d'information et anime régulièrement des conférences sur ces sujets.

Christophe Brendel – consultant senior de la Business Team e-Integration, Christophe Brendel intervient en expertise et direction de projet sur des missions EAI/SOA.

Nicolas Buche – manager de la Business Team e-Integration, Nicolas Buche intervient en expertise et direction de projet sur des missions d'urbanisme, de modélisation et de SOA/EAI.

Sebastien Delayre – consultant senior de la Business Team e-Integration et responsable de l'offre SOA d'Unilog Management, Sebastien Delayre intervient sur des études d'opportunités SOA/EAI et en tant que directeur de projet.

Alain Mocaër – senior manager de la Business Team e-Integration, Alain Mocaër mène depuis plusieurs années des missions d'urbanisation, de gouvernance du système d'information et d'accompagnement au changement pour les plus grands comptes.

Julien Nevers – consultant de la Business Team e-Integration, a participé aux parties liées aux standards de modélisation et d'exécution des processus métier.

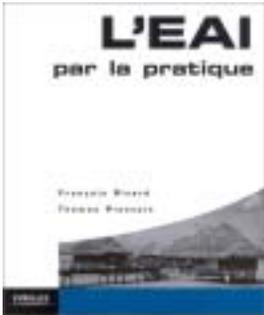
Unilog Management

Unilog Management, quatrième cabinet de conseil en France, est la branche conseil du groupe Unilog (7000 collaborateurs en Europe). Avec 800 consultants, Unilog Management a pour vocation d'accompagner les entreprises sur le conseil en management comme sur le conseil en solutions technologiques.

L'expertise développée depuis 1999 par la Business team e-Integration sur l'EAI place Unilog en première position du marché français du conseil et de l'intégration d'EAI. Cette Business team est forte de 150 managers et consultants spécialisés dans l'urbanisme et l'architecture des systèmes d'information ; Elle est constituée d'experts produit, de concepteurs et de développeurs d'architectures EAI et SOA.

Ce document a été rédigé entre mars 2004 et janvier 2005, à Bratislava, Ljubljana, L'viv, Kiev, Paris, Wrocław et Zagreb.

Les publications d'Unilog Management

	<p>Basé sur une étude de cas détaillée - le système d'information d'une agence de voyages -, cet ouvrage décrit toutes les étapes d'un projet d'intégration d'applications : de l'analyse des besoins à la modélisation fonctionnelle et technique, en passant par la sélection et la mise en place d'un produit d'EAI, le développement, le déploiement et l'exploitation. Les phases de développement et de déploiement sont illustrées à l'aide de trois des principaux outils du marché : Vitria, Tibco et webMethods. Idéal pour les chefs de projet¹.</p> <p>Prix AFISI 2003 du meilleur ouvrage informatique francophone.</p>
	<p>Edité en Juin 2003, ce livre blanc dresse un panorama de l'état d'avancement des standards liés aux Service Web et établit leur complémentarité avec les plateformes d'intégration EAI/B2B, autrefois présentées comme concurrentes, dans l'activation des Architectures Orientées Services.</p> <p>Il est illustré de références proposées par les principaux éditeurs de plate-forme d'intégration et basées sur leurs produits.</p>
	<p>Premier ouvrage en français destiné aux non spécialistes, "L'EAI au service de l'entreprise évolutive" replace l'intégration d'applications dans le contexte général de l'entreprise : les enjeux, principes, règles et méthodes dédiés à l'échange de données inter-applicatif.</p> <p>Complété d'un glossaire et d'une étude des principaux progiciels du marché, "L'EAI au service de l'entreprise évolutive" montre aux dirigeants et à leurs DSI comment structurer et optimiser les investissements technologiques passés tout en les pérennisant¹.</p>

¹ Les commentaires sont ceux du site www.indexel.net, qui place ces ouvrages parmi les TOP 5 des publications sur l'EAI.

Remerciements

Merci à toute l'équipe de la Business Team e-Intégration d'Unilog Management pour sa compétence et la qualité de ses retours d'expérience.

Unilog Management tient également à remercier :

BEA	Daniel Pacyga, Philippe Réveillon
Casewise	Jérôme Caillot, Thierry Tastet
IBM	Jérôme Carrasco, Olivier Delfosse, Jean-Pierre Rataud
Intalio	Tanguy Crusson, Jean-Louis Tournay
Microsoft	Marc Gardette, Frank Guiducci, Alexis Oger
SAP	Laurent Cozette, Patrick Janot
See Beyond	Rik De Deyn, Patrick Menahem
Sonic Software	Florent Lefebvre
Systinet	Frédéric Miszewski, Roman Stanek
Tibco	Thomas Been, François D'Haegeleer, Ghassan Khoury
WebMethods	Philippe Bessis, Régis Mauger

Table des matières

Introduction	9
Amazon.com	9
Présentation	10
1. Les Services : concepts, technologies, standards	13
1.1 Les fondations	15
1.2 Les bases de fonctionnement	16
1.3 Principes de l'approche Services	17
1.4 SOA et EDA	18
1.5 Conclusion	19
2. Organisation technologique de la SOA	21
2.1 Le tiers d'intermédiation	21
Contre le retour du spaghetti	21
L'orchestrateur SOA : du service au processus	24
Généralisation du découplage entre interface et implémentation	26
2.2 Une construction orientée Processus	28
La notation avec BPMN	28
L'exécution des processus avec BPEL	30
Le requêtage avec BPQL	31
3. Apports des SOA à l'alignement métier et l'urbanisme	32
3.1 Les outils de la rationalisation et de la réutilisabilité	32
3.2 Les apports à l'urbanisation	34
L'urbanisation itérative	34
La migration itérative des applications obsolètes	36
La construction de référentiels d'entreprise	39
L'assemblage d'applications composites	40
4. Eléments de méthode	43
4.1 Catégorisation de services	43
Du service dans toute chose	43
Verticalité de services liés aux fonctions et processus	45
Horizontalité et services transversaux	47
Cas pratique	47
4.2 L'accostage Urbanisme → SOA/EAI	49
Le contexte	49
La réponse	49
4.3 Modes de construction de la SOA	51
Cas pratique	51
Le Centre de Compétences SOA	52
Le travail sur l'existant	53
5. Les outils de la SOA	54
5.1 Le modèle technico-fonctionnel	54
5.2 Les fonctionnalités à suivre	56
5.3 Les catégories de produit	60

Les pure-players SOA.....	60
Les ESB	61
Les fournisseurs d'EAI	62
Les APS (Application Platform Suites)	63
5.4 Les acteurs et leurs produits	65
AmberPoint.....	66
BEA	68
Fiorano	71
IBM	73
Intalio.....	76
Microsoft.....	79
Oblix	81
Oracle.....	83
SAP	85
SeeBeyond.....	88
Sonic Software	91
Sterling Commerce	93
Systinet.....	95
Tibco.....	98
webMethods.....	101
6. Conclusion	103
Glossaire.....	104

Table des figures

Figure 1 – Echange entre consommateur et producteur	13
Figure 2 - Flux WSDL et SOAP dans l'utilisation de services	16
Figure 3 - Invocation de services sans utilisation d'annuaire	17
Figure 4 - Invocation directe de service à service	21
Figure 5 - Reproduction du modèle spaghetti dans l'utilisation des Services Web	22
Figure 6 - Insertion d'un tiers entre producteurs et consommateurs	23
Figure 7 – Le processus SOA : orchestration de la séquence d'appel	24
Figure 8 - Introduction d'un niveau intermédiaire : la chorégraphie	25
Figure 9 – Rôle de l'orchestrateur SOA	26
Figure 10 – Séparation entre interface et implémentation	26
Figure 11 – La dualité des couches d'abstraction	27
Figure 12 - Niveaux parallèles d'architecture	34
Figure 13 - Insertion de la couche de Services	35
Figure 14 – Cohabitation de deux applications iso-fonctionnelles	37
Figure 15 - Invocation parallèle des services	38
Figure 16 - Gestion SOA du référentiel d'entreprise	40
Figure 17 – Modèle de répartition Services/Données	41
Figure 18 - De la fonction métier au service technique	43
Figure 19 - Le modèle matriciel des SOA	44
Figure 20 - SOA et modèle de pilotage et de supervision	46
Figure 21 - Passerelles d'accostage basées sur les standards XML	50
Figure 22 - Méthode de construction itérative du SOA	52
Figure 23 - Modèle technico-fonctionnel	54
Figure 24 - Changement de version de service	58
Figure 25 - Changement de version de processus	59
Figure 26 - Les trois approches "traditionnelles" de l'EAI	62
Figure 27 – Plate-forme d'infrastructure transversale	64
Figure 28 - Modèle architectural AmberPoint	67
Figure 29 - Le modèle SOA d'Amberpoint	67
Figure 30 – Le projet Quicksilver	68
Figure 31 – Les produits de l'offre BEA	69
Figure 32 – Le modèle SOA de BEA	70
Figure 33 - L'offre de Fiorano	71
Figure 34 - Le modèle SOA de Fiorano	72
Figure 35 – Suite WebSphere	74
Figure 36 - Le modèle SOA de IBM	75
Figure 37 - n3 Designer	76
Figure 38 - Le modèle SOA d'Intalio	78
Figure 39 - Le projet Indigo de Microsoft	79
Figure 40 - Le modèle SOA de Microsoft	80
Figure 41 - le modèle SOA de Oblix	82
Figure 42 - La plate-forme APS d'Oracle	83
Figure 43 - Le modèle SOA d'Oracle	84

<i>Figure 44 - L'offre SAP Netweaver</i>	86
<i>Figure 45 - Le modèle SOA de SAP</i>	87
<i>Figure 46 - Modèle technique de SeeBeyond ESB</i>	88
<i>Figure 47 – Représentation d'un processus eInsight</i>	89
<i>Figure 48 - Le modèle SOA de SeeBeyond</i>	90
<i>Figure 49 - L'offre ESB de Sonic Software</i>	91
<i>Figure 50 – Le modèle SOA de Sonic Software</i>	92
<i>Figure 51 - Sterling Integrator</i>	93
<i>Figure 52 - Le modèle SOA de Sterling Commerce</i>	94
<i>Figure 53 - Couverture du modèle SOA par Systinet</i>	96
<i>Figure 54 - Couverture fonctionnelle et technique Systinet</i>	97
<i>Figure 55 - Positionnement de XML-Canon dans l'offre de Tibco</i>	98
<i>Figure 56 - L'environnement Business Works</i>	99
<i>Figure 57 - Le modèle SOA de Tibco</i>	100
<i>Figure 58 - Le modèle SOA de webMethods</i>	102

Introduction

Amazon.com

Il n'est pas nécessaire de présenter ici Amazon, célébrité vendeur en ligne et pionnier du B2C. Très tôt, Amazon a organisé son système d'information autour de services, mis à disposition pour ses applications en interne et/ou en externe pour ses partenaires, et dont voici une liste non exhaustive :

- Renvoi du prix et de la disponibilité d'un article en temps réel ;
- Génération à la volée d'une liste des meilleures ventes ;
- Génération d'une liste de résultat de recherche ;
- Gestion automatique des stocks ;
- Génération de rapports de commande fournisseur.

Cette liste appelle certaines remarques :

- Certains services sont destinés à être utilisés par le client Amazon depuis le site Web de la société, mais également depuis d'autres sites : l'utilisation de service permet d'exposer de façon granulaire certaines fonctionnalités à ses partenaires. C'est le cas de la liste de recherche ou de l'emplissage du caddie à distance depuis un autre site : on expose des éléments d'une application et non l'application elle-même, en fournissant un accès à certaines fonctionnalités clairement référencées par les services qu'elles rendent. Cette notion de référencement ouvre la voie à celle **d'annuaire de service**, nœud central qui informe des modes d'accès au service et de ses interfaces d'entrée-sortie.
- L'idée est ainsi de permettre à tout utilisateur, à toute application, d'accéder aux services qui lui sont nécessaires pour bâtir ses propres processus. Amazon devient fournisseur de services pour lui-même (en interne) comme pour ses partenaires. Le service est réalisé une seule fois et potentiellement utilisable par tout partenaire disposant des bonnes habilitations. Comme dans toute logique B2B, l'exposition de services ne fait pas l'économie d'une politique de sécurité judicieuse et précise.
- Certains services ne sont destinés qu'à des services internes de Amazon à des fins de bonne gestion. Ils peuvent être exposés en interne à d'autres applications, de façon autonome ou comme élément d'un processus métier plus vaste ; ils peuvent également être utilisés par plusieurs processus d'une même entreprise ou d'entreprises différentes. Réutilisabilité et mutualisation des composants sont deux mots-clés des architectures à base de services, qui légitiment l'utilisation de plusieurs outils pour garder la main sur le système construit. L'utilisation d'un annuaire en est un (c'est le cas chez Amazon avec le produit de Systinet), mais, pour construire un système qui généralise l'utilisation des services à un niveau d'entreprise, ce ne peut être le seul.
- Cet exemple fournit un bon contexte d'utilisation des services : poussée par la réactivité et le besoin de s'adapter à un environnement changeant, l'entreprise met en place une infrastructure de service qui lui permet notamment d'inclure le temps réel dans bon nombre de ses procédures client. Ici, le temps réel est une priorité, la pierre fondatrice de

tout le modèle. La capacité adaptative étant par ailleurs l'apanage d'autres infrastructures, celles à base d'EAI, on voit apparaître des points de convergence et de divergence entre ces deux modèles. Ce point sera largement approfondi dans la suite de ce document.

Présentation

Cet exemple témoigne concrètement du développement de l'un des domaines de l'industrie informatique parmi les plus prometteurs et les plus enthousiasmants de ces trois dernières années : les Services Web et les Architectures Orientées Service (SOA pour *Services Oriented Architecture*). Il montre également qu'il en existe quelques implémentations industrielles, dans des contextes d'entreprise réclamant une qualité de service irréprochable (tenue à la charge, robustesse et haute disponibilité), même si nous ne sommes pas encore entrés dans une réelle phase de généralisation de ces méthodes à l'ensemble des systèmes d'information des grandes entreprises. En France, le cap des *early adapters* a cependant été dépassé, de même que le cadre de projets pilotes de peu d'envergure, comme en témoignent les quelques cas client présentés dans ce livre blanc.

Ce tournant important est actuellement pris par une majorité d'acteurs de l'industrie, grands comme petits, spécialistes comme généralistes, éditeurs comme cabinets de conseil, SSII ou entreprise utilisatrice, et il ne semble pas qu'une quelconque marche arrière puisse être envisagée. Nombreux sont les éditeurs de logiciels qui réarchitecturent leur offre produit vers l'assemblage de services, et ce, quelle qu'en soit la nature. Nombreuses sont également les entreprises séduites par les promesses affichées de meilleure gouvernance de leur système d'information, d'interopérabilité globale, et par la possibilité de mettre à disposition un service unique pour de nombreux acteurs, internes comme externes à l'entreprise.

Par contre, il est important de bien mesurer les enjeux et de prendre le train du bon pied, pour tirer le maximum de ce que ces concepts et technologies promettent. Se poser les bonnes questions et apporter les bonnes réponses dès aujourd'hui permettra aux systèmes d'informations construits sur ce modèle de durer dans le temps et de fournir un socle stable de construction des systèmes et des applications.

Cette effervescence témoigne de la capacité du couple Web Services/SOA à répondre à de nombreux enjeux auxquels fait face l'entreprise pour demeurer performante et continuer à progresser. On le voit au travers des exemples présentés ci-dessus : les SOA apportent un second souffle à des travaux d'urbanisation de système et de cartographie qui souffrent parfois d'un préjudiciable manque de pragmatisme. L'alignement métier et la rationalisation des systèmes informatiques sont devenus des priorités pour de nombreux DSI et les concepts et technologies que nous allons présenter en détail dans ce document facilitent la capacité de réponse à ces enjeux. Grâce à ces concepts et à ces technologies, la conception et/ou le remodelage de systèmes d'information, la construction d'applications, la constitution de référentiels d'entreprise, s'accompagnent plus aisément d'une réflexion fonctionnelle et urbanistique qui renforce les capacités d'alignement métier de l'IT.

Nous chercherons cependant à ne pas tomber dans un enthousiasme ou un triomphalisme béat. En effet, on adore, dans l'industrie informatique, faire du neuf avec du vieux, et resservir à l'envi de vieilles scies éculées, moyennant un lifting de circonstance. On peut donc s'interroger si cette

omniprésente notion de Services n'est pas, peu ou prou, autre chose qu'un retour sous un autre nom des bus de composants ou autres architectures client/services (le client/serveur dit de 4^{ème} génération) sur le devant de la scène (c'est par ailleurs une question à laquelle nous avons souvent à répondre aujourd'hui lorsque nous venons présenter les SOA). Par certains côtés, il est vrai que le modèle d'architecture sous-jacent est largement inspiré de ce qu'a proposé en son temps DCE ou CORBA par exemple, sans parvenir pour autant à s'imposer universellement. Les SOA se proposent d'étendre et d'améliorer ces modèles, tant sur le plan fonctionnel que technologique, par le jeu d'innovations largement industrialisées depuis : les technologies de l'Internet, qui ont considérablement rajeuni la notion de composant, d'architecture distribuée et de bus de services :

- en lui conférant une teinte fonctionnelle bien plus marquée,
- en standardisant les protocoles utilisés pour aboutir à une véritable promesse d'interopérabilité technique, à moindre frais, pour tout système,
- et en fragmentant l'application, bloc relativement monolithique de fonctions, pour lui apporter une granularité alléchante dans le contexte généralisé d'urbanisation, que l'on retrouve actuellement dans de nombreuses entreprises.

Aussi, il convient de prendre un peu de recul, de donner quelques pistes et de répondre aux questions qui se posent concernant les points les moins clairs :

- **Les services** : que recouvre précisément cette notion ? Comment cela fonctionne-t-il ? Quels sont les types et les catégories de services ?
- **Normes et état d'avancement des standards** : devant la prolifération de standards, quelles sont les initiatives dignes d'intérêt ? Lesquelles structurent aujourd'hui et vont structurer demain les SOA ? Quel est leur degré de maturité ?
- **SOA et architecture** : quels logiciels, quelles technologies, quels niveaux de services participent à l'organisation des SOA ?
- **SOA et urbanisme** : quel est l'apport réel de ces concepts/technologies à l'alignement métier des systèmes d'information ? A quel niveau de granularité, à quelle maille métier, se situent réellement les services ?
- **SOA, APS, EAI et ESB** : on entend beaucoup parler de la concurrence que se livrent les technologies d'instrumentalisation de SOA, notamment les APS (Application Platform Suite), ESB (Enterprise Service Bus), et plates-formes d'intégration EAI (Enterprise Application Integration). Nous dissiperons ici tous les malentendus et dresserons un point précis de la situation.
- **Produits** : quels types de produits supportent la mise en œuvre d'une SOA ? Quelles fonctionnalités doit-on y trouver ? Quel niveau d'implémentation des standards y trouve-t-on ?
- **Méthode** : comment s'y prendre pour construire une SOA dans le cadre d'une démarche d'entreprise ?

Ce document est structuré en suivant chacun des points cités ci-dessus, à savoir :

- une première partie qui présente et identifie la notion de services, tout en opérant un lien avec les Services Web et leur rôle dans les Architectures Orientées Services,
- une seconde partie qui présente certaines normes et technologies et explique les raisons qui poussent à les utiliser lors de la mise en œuvre d'une SOA,

- une troisième partie qui fait la part belle aux réponses qu'apportent les SOA à quelques grandes questions urbanistiques,
- une quatrième partie qui présente des éléments de méthode, notamment l'accostage Urbanisme $\leftarrow \rightarrow$ SOA, indispensable à l'exploitation et à la maintenance réussie d'une SOA,
- et enfin, une cinquième partie où sont évoquées les catégories de produits capables d'outiller la démarche, ainsi que les principaux éditeurs, visions et produit du marché,

1. SOA et services : concepts, technologies, standards

Que recouvre précisément la notion de Services ? Principalement, on y trouve une redéfinition des applications, qui se réorganisent en ensembles fonctionnels dénommés services, dont certains sont exposés au monde extérieur, en tant que *producteurs* de services, pour des *consommateurs* de services, eux-mêmes services ou applications.

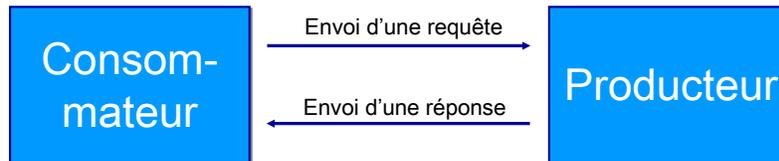


Figure 1 – Echange entre consommateur et producteur

Construire une architecture à base de services revient ainsi à organiser l'agencement, les séquences, les droits d'accès... à ces modules, composants, entités fonctionnelles, mais aussi techniques, capables d'assurer une fonction et d'en rendre compte. Cela peut sembler trivial, mais encore faut-il reconnaître que tous les systèmes informatiques ne se reconnaissent pas dans cette définition. Certains ne sont pas prêts. De façon caricaturale peut-on même estimer qu'une application monolithique ne sera guère capable que d'exposer un service unique : elle-même... Services et SOA risquent d'y perdre une bonne partie de leur charme. L'exposition de services et la mise en œuvre de SOA n'est-elle possible que pour certaines entreprises qui ont eu le flair de s'entourer par le passé des technologies adéquates ? Toutes ne pourraient-elles pas en profiter ?

Fort heureusement, il n'en est pas si souvent ainsi ; il existe des techniques pour segmenter une application en fonctions, et les exposer sous forme de services, même si la granularité de ces services reste variable en fonction de l'application considérée. L'existant technologique actuel présente ainsi quelques contraintes dans la liberté de définir un système à la carte, mais cette situation va se résorber dans le temps, avec l'évolution des technologies et le renouvellement des parcs applicatifs des entreprises. La construction d'une première série de services (services communs d'accès aux données, production, traitement des flux, production des éditions, authentification, cryptage,...) permettra déjà de défricher les possibilités et contraintes. Certes, il restera du chemin à franchir pour transformer les blocs applicatifs obtenus en véritables composants de services fonctionnels (services métiers) au sens SOA du terme, mais une telle opération est tout à fait réaliste, au moins pour une partie des applications concernées.

Cette évolution peut par ailleurs être accélérée au travers de la mise en œuvre au préalable d'une démarche d'urbanisation d'applications (désimbrication fonctionnelle, modularisation, externalisation, mutualisation des services communs (données, échanges, éditions, sécurité). Cela permet de transformer progressivement le parc applicatif en construisant des blocs plus fins, ciblés sur des macro-fonctions du système, plus faciles à intégrer (y compris dans une architecture hétérogène répartie) et à faire évoluer (refonte, remplacement).

Présenter la notion de service revient également à évoquer les liens qu'entretiennent le concept de Services avec les technologies des Web Services ? On estime généralement que bâtir des Web Services n'est qu'une façon de construire une SOA, partielle ou d'entreprise. Celle-ci est particulièrement intéressante en regard des promesses d'interopérabilité qu'elle affiche, mais

peut parfaitement cohabiter, au sein d'une SOA, avec d'autres modes d'implémentation. A l'extrême, on pourrait même envisager des architectures où ne figurerait aucun Web Service.

Cependant, si l'on veut procéder de façon chronologique, il est bon de se souvenir que la notion de SOA fut d'abord précédée de l'émergence des Services Web, et c'est dans leur sillon que celle-ci prit ensuite corps et s'affirma. En effet, la promesse d'interopérabilité apportée par les Web Services apportait la concrétisation technologique de concepts déjà présents et/ou latents, mais en les renforçant. Finalement, on retrouve les principaux concepts des SOA dans les principes fondateurs de CORBA ou d'autres technologies de bus applicatifs organisées autour de l'invocation distante de méthodes. On est en droit de se poser la question : pourquoi les SOA réussiraient-ils là où CORBA, modèle universel, n'a pas réussi à s'imposer ? Peut-être précisément parce que l'actuelle implémentation de référence des SOA, les Web Services, apportent des réponses qui n'existaient pas auparavant. Leur mérite réside dans la standardisation de l'accès aux Services pour tout le système d'information, celle par laquelle tous les acteurs de l'industrie du logiciel, grands comme petits, établissent et établiront demain l'accès principal aux fonctions de leurs applicatifs. De plus, là où les modèles proposés étaient initialement synchrones, les outils d'activation des SOA apportent une notion d'asynchronisme (également nommé EDA pour Event-Driven Architecture) qui répond aux enjeux récurrents de flexibilité en s'inspirant des modèles technologiques du monde de l'EAI.

Massivement adoptée par tous ces acteurs, les Services Web ne peuvent que massivement se généraliser en entreprise. Il faut cependant se garder de tout optimisme exagéré. L'exemple d'Amazon nous rappelle qu'au-delà des standards d'invocation à distance de méthodes, les architectures à base de Services Web doivent proposer des langages complémentaires pour la gestion des transactions ou de la sécurité. Le niveau de maturité et de stabilité de ces langages reste assez variable, même si certains, comme WS-Security, commencent à être largement implémentés dans les produits ; mais ces chantiers nous rappellent que le modèle proposé reste encore assez loin des services que propose, par exemple, un bus CORBA. Et d'autres initiatives peuvent laisser sceptiques, voire inquiéter : ainsi, pourquoi lancer un groupe de travail WS-I Basic Profile, dont la vocation est de fournir des spécifications aux éditeurs, en vue de maintenir l'interopérabilité des standards !

Ainsi, on constate, que les standards des Web Services restent indissociables d'une présentation des SOA, et il est important de se pencher sur l'état d'avancement de ces travaux pour comprendre ceux sur lesquels les entreprises peuvent et doivent d'ores et déjà s'appuyer, et ceux qui ne restent encore que de prometteuses perspectives, mais qu'il convient de surveiller de près.

1.1 Les fondations

Dans les pages qui vont suivre, nous allons inévitablement nous retrouver confrontés à un ensemble d'acronymes dont est friande l'industrie informatique. Vous les retrouverez regroupés dans le glossaire que nous avons mis à votre disposition à la fin de ce document. Cependant, il nous paraît important de présenter de façon plus détaillée les principaux termes que nous reprenons immédiatement, même si certains d'entre eux sont désormais célèbres.

Standard	Description
SOAP	Simple Object Access Protocol Langage XML standardisé de requête/réponse avec lequel invoquer les services. Utilisé pour encapsuler les données qui permettent au service de s'exécuter ainsi que les données retournées par le service.
WSDL	Web Services Description Language Langage XML standardisé de description des interfaces de service. Utilisé pour connaître les modalités techniques d'accès à un service, le nom des fonctions qu'il expose, les données qu'il attend pour s'exécuter et les données qu'il renvoie en retour.
UDDI	Universal Description, Discovery & Integration Technologie de structuration d'annuaire dans le but de fournir une description standardisée des services référencés par celui-ci, et permettre la découverte dynamique de ces services.

UDDI est une technologie de structuration d'annuaire, alors que WSDL et SOAP sont des protocoles utilisés pour l'échange d'information avec les services ou au sujet de ceux-ci. L'utilisation complémentaire de ces trois standards est schématisée à la Figure 2 page 16.

Si SOAP et WSDL sont désormais stables, UDDI peine encore à faire l'unanimité, notamment dans sa gestion de la sécurité et des aspects contractuels qui doivent faciliter la découverte dynamique des services, et il reste difficile de prédire la pérennité des implémentations existantes, basées sur des versions intermédiaires, lorsqu'il faudra les mettre en conformité avec la version définitive de la spécification.

1.2 Les bases de fonctionnement

L'imbrication de ces trois normes dans la découverte et l'utilisation de services peut se schématiser dans le diagramme de séquence suivant.

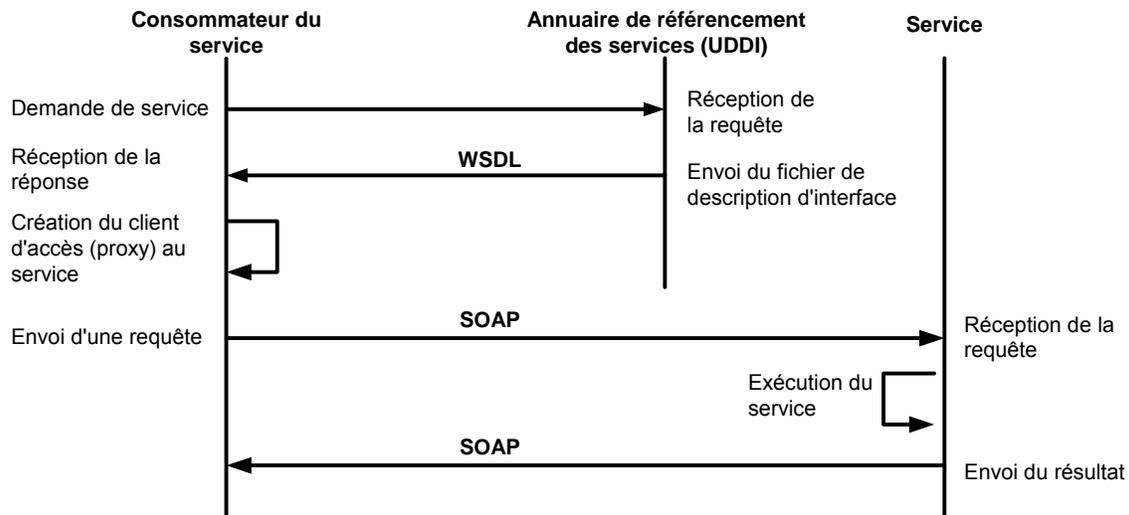


Figure 2 - Flux WSDL et SOAP dans l'utilisation de services

L'annuaire UDDI stocke les descriptions d'interface de services et les met à disposition des consommateurs qui en font la demande. L'annuaire est présent en tant que point d'accès central pour tout consommateur, et c'est grâce à lui qu'un consommateur découvrira non seulement la structure des fonctions exposées par le service, mais également l'emplacement physique de celui-ci.

C'est, normalement, également grâce à l'annuaire qu'une SOA peut assurer les mécanismes de découverte dynamique de services, ce qui prête davantage matière à discussion, et nous en débattons par ailleurs dans la prochaine partie de ce document.

Les premiers échanges entre l'annuaire et le futur consommateur de service s'opèrent sur la base d'une requête de document WSDL, elle-même empaquetée dans un document SOAP. L'annuaire remet le document WSDL au consommateur, qui crée localement une fonction nécessaire à l'invocation distante du service. Cette opération effectuée, l'invocation du service est possible : tout appel d'une des fonctions du service déclenche la génération d'un message SOAP, envoyé directement au service, qui renvoie en retour le résultat de son exécution au consommateur. On ne peut donc pas encore réellement parler pour l'instant de découplage entre consommateur et service, puisque le consommateur fait directement appel au service. L'annuaire n'est utilisé que pour accéder à la description et à l'interface du service. Nous verrons au chapitre 2 comment la construction d'une véritable SOA, avec un organisateur des échanges entre les services et leurs consommateurs, permet de contourner ces limites.

Pour l'instant, livrons-nous à notre première réflexion d'architecture, et concentrons-nous sur le visage qu'aurait l'invocation de services sans annuaire. Il suivrait probablement la séquence suivante :

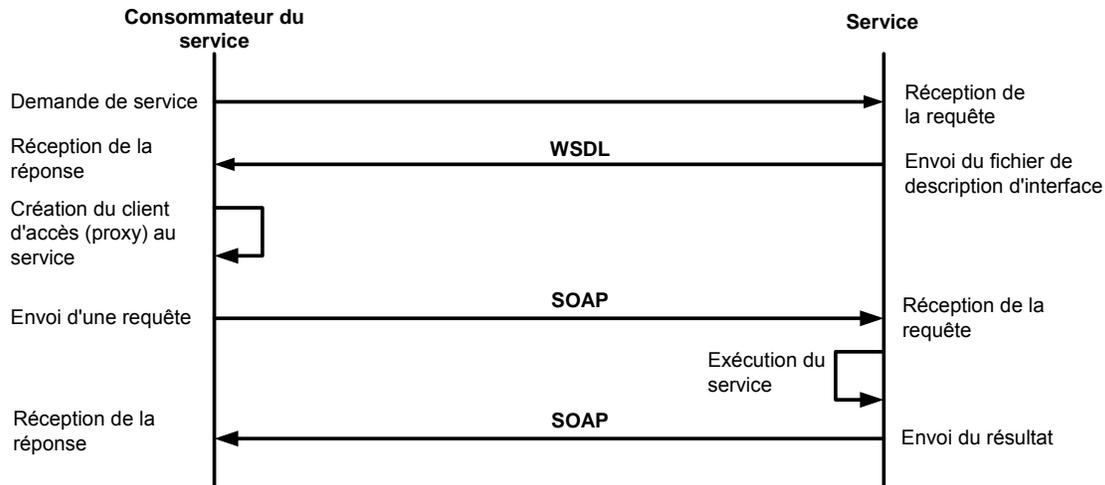


Figure 3 - Invocation de services sans utilisation d'annuaire

Dans ce schéma, il incombe au consommateur du service de connaître l'emplacement physique des services qu'il invoque. On comprend immédiatement combien cette situation est génératrice de rigidité et doit être évitée. On en déduit la nécessité de placer un tiers médiateur entre les services et leurs consommateurs. Ce tiers est, dans notre exemple, un annuaire UDDI. Nous verrons dans la partie consacrée aux produits les différentes formes que ce tiers peut en réalité emprunter.

Dans tous les cas, les trois standards présentés forment l'ossature de base du modèle sur lequel reposent les Services Web. Nous n'entrerons pas ici dans une présentation détaillée de ces normes, de même que nous ne nous étendrons pas davantage sur la galaxie des standards qui ont vu le jour autour des Web Services pour structurer des aspects aussi divers que le routage de données, la sécurité, les transactions².

1.3 Principes de l'approche Services

Pour faire de cet encourageant tableau une réalité, la notion de Service repose sur quatre fondamentaux :

- **Support d'interfaces basées sur les standards du marché** : au-delà de SOAP, d'autres protocoles sont standardisés ou en voie de standardisation dans le monde des Web Services. Plusieurs groupes de travail existent, constitués de nombreuses sociétés industrielles, et non des moindres puisque les plus visibles sont BEA, IBM et Microsoft. Les couches basses (SOAP, WSDL, UDDI) du modèle sont stabilisées. Les couches transversales (transactions, sécurité...) sont en partie stabilisées et implémentées, en partie en cours de stabilisation. Les langages de description des processus (BPMN, BPEL) sont dans un bon état d'avancement et désormais implémentés dans les produits du marché.
- **Séparation de l'interface et de l'implémentation** : tout ce que l'on doit connaître du service, en dehors de sa propre mission, sont ses interfaces d'entrée et de sortie, c'est-à-dire

² Nous vous renvoyons pour cela au livre blanc « Les Web Services dans l'EAI/B2B » - J-M. Zuliani, G. Abou Harb – Juin 2003 – Unilog Management

les informations qu'il attend pour s'exécuter, le résultat qu'il produit et les moyens technologiques, comme les protocoles utilisables, pour y accéder. Le contrat d'interface établi entre les consommateurs du service et le service lui-même est intangible et ne doit en aucun cas être modifié pour une même version d'un service. La façon dont le service s'exécute et les règles métier qu'il applique forment une boîte noire.

- **Neutralité technologique** : la technologie avec laquelle sont implémentés les services importe peu, puisque l'accès au service s'opère depuis une interface mise à disposition sur SOAP, protocole normalisé. La promesse d'interopérabilité qui en résulte est un des principaux facteurs de l'engouement envers les architectures orientées services. L'accès peut ensuite s'opérer sur http, sur SMTP, JMS ou tout autre protocole, l'idée étant d'aller vers un découplage de la description de l'interface d'entrée/sortie (fonctionnelle) et de la description des protocoles d'accès (techniques). Certains éditeurs, tels qu'IBM, génèrent ainsi des fichiers WSDL qui séparent ces deux parties et vont dans le sens du découplage entre le métier et l'infrastructure.
- **Découverte dynamique du producteur par le consommateur** : ces fonctionnalités de découverte dynamique des services pour le consommateur firent les beaux jours des zéloteurs des Web Services, et principalement de ceux qui y voyaient un apport supplémentaire de flexibilité pour l'entreprise étendue : il devenait subitement particulièrement simple d'obtenir à la carte, à la demande, via les services d'annuaire, un service moins onéreux ou plus performant. Mais, cela revenant non pas uniquement à changer de version de service, mais plutôt de partenaire B2B, on s'aperçoit que l'opération n'est pas si simple. Si, techniquement, elle semble réalisable (quoique les interfaces d'accès aux services ne sont pas normalisées), le jeu des contrats et des abonnements entre partenaire rigidifie singulièrement le modèle technique. On ne souscrit pas à un service externe sans notification ou sans négociation préalable. Ainsi, cette fonctionnalité de découverte dynamique est-elle encore loin d'être applicable et automatisée. Elle ne constitue pas cependant, et de loin, l'enjeu majeur à l'heure actuelle de la construction d'une SOA ou de l'adoption des Web Services.

1.4 SOA et EDA

Pour conclure avec les définitions, il est de coutume d'opposer la SOA, synchrone, à l'EDA (Event-Driven Architecture, Architecture Pilotée par les Evènements), asynchrone. Sans vouloir tomber à tout prix dans le contre-pied permanent, il faut pourtant bien constater que les deux notions ne sont pas de périmètre égal et forment ainsi un ensemble plutôt complémentaire.

Généralement, l'idée sous-jacente est qu'une SOA serait pilotée par les consommateurs de services, de façon assez mécanique, suivant une fréquence fixe et un mode requête/réponse indéfectiblement synchrone, imposant un couplage serré. La SOA serait ainsi l'architecture permettant d'organiser au mieux, la découverte et la communication entre les consommateurs et les producteurs, mais elle créerait des dépendances fortes. Ces dépendances doivent être évitées : l'adoption du mode asynchrone s'impose pour permettre un couplage lâche.

D'autre part, certains services ou enchaînements de services ne seraient pas toujours déclenchés à l'initiative d'un consommateur, mais pourrait être déclenchés par un évènement métier ou technique de fréquence variable. C'est ainsi qu'est arrivée sur le devant de la scène la

notion d'EDA, susceptible de gérer l'évènementiel. On voit que cette notion répond à des lacunes du modèle SOA, ceci afin de l'enrichir, et non de s'y opposer ou de proposer un modèle alternatif.

Ainsi, le véritable modèle SOA serait celui qui disposerait également de cette indispensable gestion évènementielle et asynchrone. A titre de comparaison, les premières spécifications de la norme JCA (*Java Connector Architecture*) étaient uniquement synchrones et mono-directionnelles, initiés par le serveur d'applications. Cela n'était pas suffisant pour permettre aux serveurs d'application de s'opposer alors aux produits d'EAI, qui proposaient des modèles évènementiels asynchrones, où l'initiation d'un processus pouvait venir de tout type d'application ou d'acteur. Cela s'est traduit finalement par une complémentarité entre les deux types d'outils, qui les voit se rejoindre et partager les rôles au sein de plates-formes d'infrastructure élargies :

- Pour l'EAI, un orchestrateur évènementiel de processus transversaux technico-fonctionnels et un middleware asynchrone orienté messages ;
- Pour le serveur d'applications, l'organisation de l'exécution de composants métier.

Comme nous aurons de le voir dans la suite de ce document, l'organisation d'une SOA se construira sur cette même dualité.

1.5 Conclusion

Une SOA couvre ainsi les modes de conception et d'organisation opérationnelle d'un système d'information pensé pour délivrer des services à tous les consommateurs qui en ont le besoin. La notion d'annuaire devient centrale et incontournable. Elle permet aux consommateurs de trouver rapidement et efficacement leur bonheur dans un système qui va croître en richesse mais aussi en complexité avec le temps. Si le composant recherché n'existe pas, les consommateurs pourront en faire la demande auprès d'un Centre de Compétences dûment habilité, en suivant des procédures collaboratives de demande et d'approbation. Au-delà de l'aspect purement technologique et des moyens utilisés, l'organisation collaborative d'une SOA et le dimensionnement d'un Centre de Compétences sont les clés de la réussite et de la pérennité de l'initiative dans l'entreprise.

Tous les éléments présentés sont tournés vers le même objectif de rationalisation du système d'information, par une réutilisabilité accrue des services réalisés : on gère et on administre l'ensemble des services présents dans le système. Si on ne peut pas mutualiser à l'infini un composant, on est au moins informé de sa duplication, sachant que moins il y aura de composants dupliqués dans un système d'information, plus ces composants seront faciles à maintenir et à faire évoluer, et moins coûteuse en sera la maintenance. L'administration d'une SOA devient un sujet d'arbitrage et de compromis, de règles d'urbanisation à appliquer avec rigueur, pour maintenir une direction, mais sans se plier pour autant à un dogme outrancier, aux effets destructeurs, en conservant souplesse et flexibilité.

L'objectif de réutilisabilité s'accompagne d'un autre enjeu qui n'apparaît qu'en filigrane : construire et préserver la flexibilité du système d'information. Une SOA mal organisée peut potentiellement créer des situations anarchiques, à la complexité croissante, qui ruineront les efforts entrepris. Il convient dès lors de se doter des bons outils et des bonnes techniques qui permettront au DSI et à ses architectes de se prémunir contre ces situations. Ce faisant, la situation d'ensemble du SI sera plus simple à lire, à maîtriser, et le SI en sera d'autant plus gouvernable. Toute volonté d'alignement métier sera plus facile et plus rapide à concrétiser.

Nous allons évoquer ces aspects dans la partie qui suit, consacrée à l'organisation technologique de la SOA.

2. Organisation technologique de la SOA

L'organisation technologique des SOA repose sur deux éléments phares :

- Le positionnement dans l'architecture d'un tiers d'intermédiation, qui orchestrera dynamiquement le modèle, sur un plan métier comme sur un plan technique, et assurera un couplage lâche entre les services pour fournir une moelle épinière, un véritable système nerveux d'entreprise ;
- L'adoption d'un bouquet de standards, dont certains restent en cours de finalisation, tournés autour du processus métier.

2.1 Le tiers d'intermédiation

Contre le retour du spaghetti

Rationalisation et réutilisabilité ne resteront jamais que de vains mots si on ne prête attention, dès les premiers pas de la démarche, à un piège dans lequel on peut tomber à tout moment, et qui va jusqu'à remettre en cause certains grands principes pourtant communément considérés comme acquis. L'un des ceux-ci est la volonté de délivrer le système d'information du fameux syndrome spaghetti, qui se manifeste par un système informatique où les applications sont fortement couplées entre elles, par le biais d'interfaces d'échanges souvent construites sur des technologies hétérogènes.

Cette volonté de découpler des applications, dont l'imbrication constitue un frein à la nécessaire évolution des systèmes d'information, est un des moteurs qui ont conduit à l'émergence et à l'essor des plate-forme d'intégration EAI et à les positionner comme des leviers de flexibilité et d'agilité pour le système d'information, élevant finalement ces produits au rang d'instruments de premier ordre dans la conduite d'une politique d'urbanisation. Or, la question du découplage, ou en tout cas du couplage lâche, se pose de nouveau dans le contexte des SOA, où l'on cherche à établir des relations, statiques ou dynamiques, entre les services et leurs consommateurs potentiels.

Or, par conversation directe de service à service, sans outil tiers pour organiser cette conversation, on recrée un couplage serré, direct, de service à service, comme le montre la figure ci-dessous.

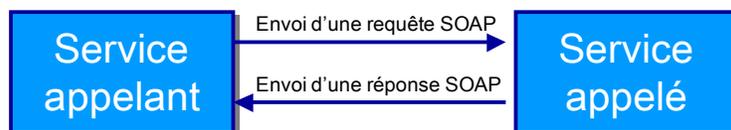


Figure 4 - Invocation directe de service à service

Or, tous ceux qui se sont intéressés de près ou de loin aux concepts de l'EAI connaissent tous les maux générés par l'échange point-à-point : ceux-ci sont, tant que faire se peut, à proscrire : en créant des passerelles directes entre les applications, on finit par figer le système d'information dans son ensemble, avec des conséquences néfastes pour l'évolutivité et la flexibilité de l'ensemble.

Si ce système de dialogue direct de la figure se généralise, on aboutit inévitablement à reproduire le modèle spaghetti, comme le montre la figure 4. Ce retour en arrière guette clairement tout SOA non organisé³.

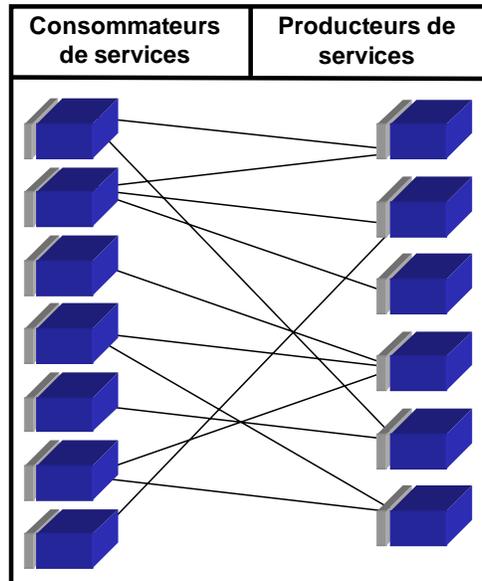


Figure 5 - Reproduction du modèle spaghetti dans l'utilisation des Services Web

Que l'échange d'information s'opère entre applications ou, comme ci-dessus, entre services, on va chercher à privilégier des solutions qui favorisent un couplage lâche entre les composants. L'introduction d'un composant intermédiaire, que nous nommerons dans un premier temps le tiers d'intermédiation, rationalise les échanges et découple les services, comme le montre la ci-dessous ci-dessus.

³ Notons que dans ce schéma, nous n'opérons pas de distinction entre le Producteur du Service et le Service lui-même, qu'il est d'usage de distinguer. En effet, il faudrait en réalité dissocier les deux, le Service étant normalement un « habillage » du producteur de Service. Nous avons préféré unifier les représentations, pour faciliter la lecture et la compréhension dans un premier temps.

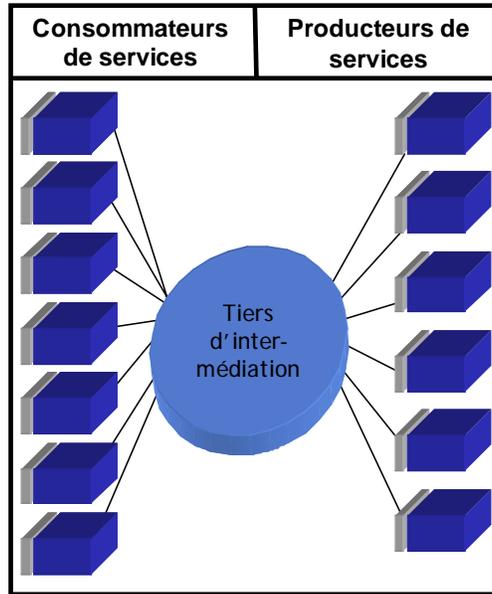


Figure 6 - Insertion d'un tiers entre producteurs et consommateurs

En tant que tiers d'intermédiation, l'utilisation d'un annuaire de services seul, de type UDDI par exemple, peut ne pas suffire s'il n'est utilisé qu'à des fins de découverte du producteur par le consommateur, par distribution des fichiers de description WSDL. Fort heureusement, cette utilisation de l'annuaire, somme toute limitative puisque réservée au seul processus de découverte, tend à se réduire au profit d'une utilisation plus opérationnelle, où l'annuaire se charge de distribuer les invocations et les réponses et d'organiser ainsi les communications de service à service. Dans ce domaine, l'offre de l'éditeur américain Systinet propose des produits qui respectent totalement ce périmètre d'utilisation.

Pour reprendre un point évoqué dans la première partie de ce document, on va également utiliser ce tiers d'intermédiation pour couvrir les indispensables fonctionnalités d'annuaire et permettre aux services de se découvrir et de se référencer. Mais au-delà, de par sa position dans le système, le tiers d'intermédiation devra couvrir encore d'autres responsabilités transversales indispensables au bon fonctionnement et à la performance du système, parmi lesquelles :

- Une garantie de réponse et de réponse unique à une sollicitation de la part d'un consommateur de service. Cela va se notamment se traduire par l'introduction de technologie de transport asynchrone, comme il est coutume de les rencontrer dans les sujets liés à l'intégration EAI.
- Une capacité de montée en charge et de tolérance aux pannes, assurée par une distribution de l'annuaire sur tous les points du réseau (ou nœuds) où le système est activé
- Une gestion de la sécurité des données et des échanges (chiffrement, certification, non-répudiation...), notamment dans un contexte d'échange avec des partenaires, les services pouvant, ainsi que nous l'avons également vu, être exposés au monde extérieur.

L'orchestrateur SOA : du service au processus

Il est temps de donner un nom à notre tiers d'intermédiation, et nous le nommerons Orchestrateur SOA. Cette idée d'orchestration élargit le périmètre d'intervention de ce tiers : il n'est pas uniquement présent pour assurer une communication sécurisée et un couplage lâche entre consommateurs et producteurs de services. Il est également présent pour organiser et séquencer les appels de services dans des processus techniques ou métier, en réponse à des événements ou de façon programmée.

La seconde étape consiste à intégrer ensemble des services pour accomplir une tâche ou implémenter un processus métier. Le processus est lui-même un service et peut être déployé et distribué en plusieurs points physiques et accessibles séparément.

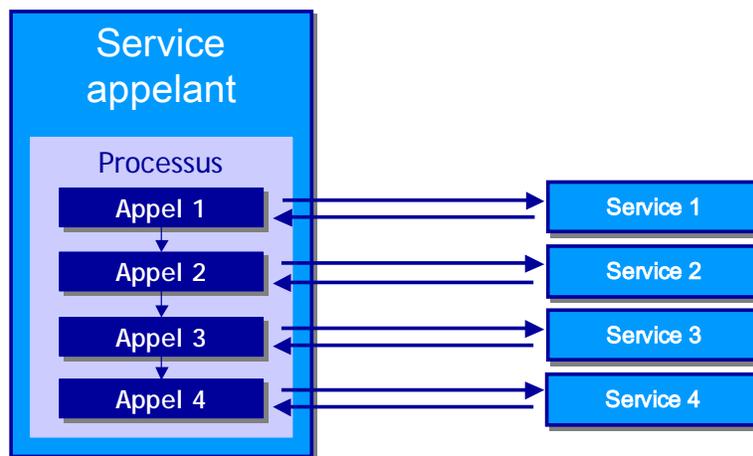


Figure 7 – Le processus SOA : orchestration de la séquence d'appel

Pourtant, au plus bas niveau, la granularité du service est souvent technique, et donc trop faible pour décrire une fonction métier, alors qu'au plus haut niveau, celui du processus, on se situe souvent à un niveau de description trop haut, qui demande à être segmenté. Il est donc nécessaire, dans une troisième étape, de généraliser ce modèle d'organisation en travaillant sur la granularisation du SI, de deux façons

- en introduisant autant de niveaux intermédiaires que l'exige l'organisation des services et des fonctions, et notamment en pré-assemblant des blocs de services techniques unitaires selon des séquences techniquement et fonctionnellement cohérentes, qui décrivent ce que l'on nomme des « chorégraphies », ainsi que le montre la figure 5 ci-dessous.

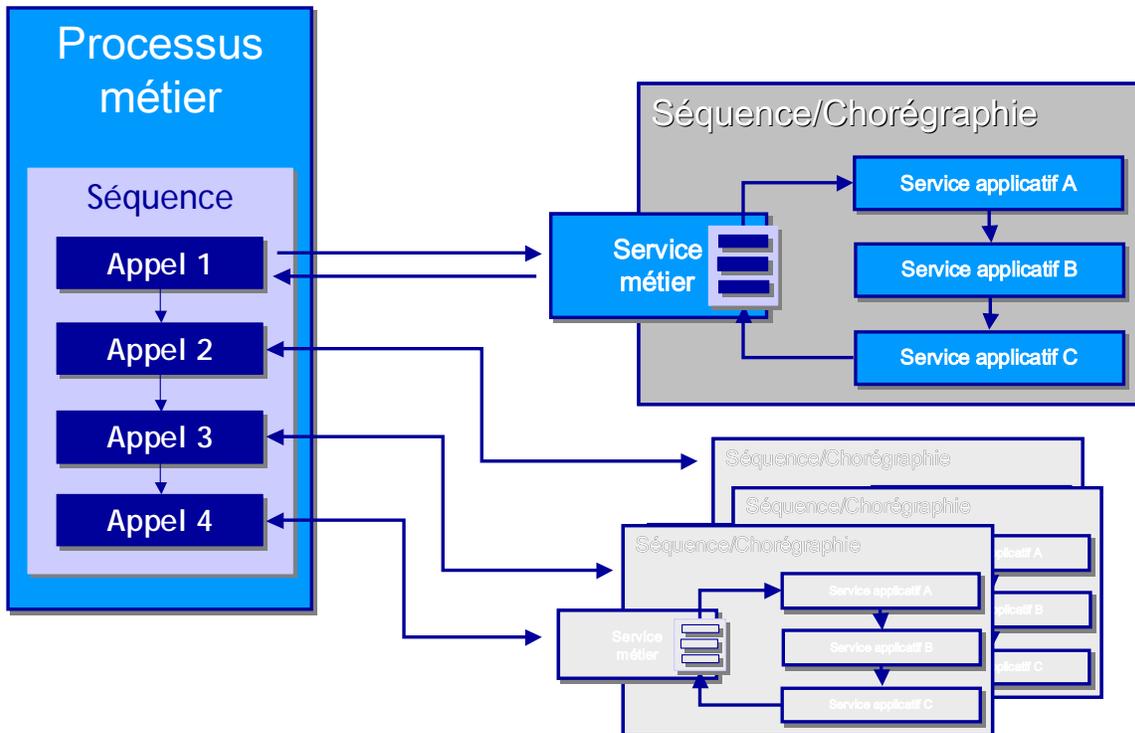


Figure 8 - Introduction d'un niveau intermédiaire : la chorégraphie

L'assemblage de chorégraphies apporte un premier niveau de sens à des services métier, par exemple par l'introduction d'un contexte transactionnel.

En reproduisant un modèle de découpage successif de granularité forte à fine, entre processus, procédures, chorégraphie et services, qui propose de plus une dissociation conforme aux principes des Services entre interface et implémentation. Un modèle particulièrement abouti de cette dissociation, au niveau fonctionnel comme technique, est détaillé dans la présentation (plus loin dans ce document) du produit n³ de l'éditeur Intalio.

Tout appel de service étant susceptible de déclencher des appels en cascade et séquentiels d'autres services, le rôle de l'orchestrateur SOA sera de gérer ces enchaînements de services imbriqués de la meilleure façon possible. A différents niveaux, les processus pourront être directement modélisés et implémentés depuis l'orchestrateur, qui se chargera d'invoquer les services applicatifs correspondants, ainsi que le montre la figure 9 ci-dessous.

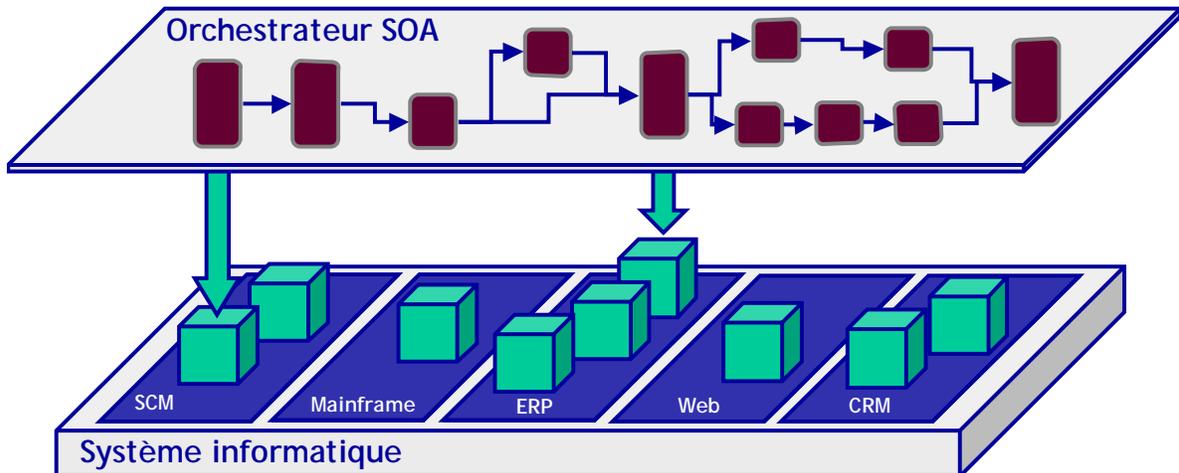


Figure 9 – Rôle de l'orchestrateur SOA

L'orchestrateur SOA doit être capable de couvrir les besoins liés à la gestion de ces séquences métier complexes, telles que les transactions, les interventions d'un utilisateur dans le cours du processus, les conditions et les boucles, les états longs suspendus... Le processus ainsi créé pourra lui-même être exposé sous forme de service.

Généralisation du découplage entre interface et implémentation

Le modèle d'orchestration SOA ci-dessus ne serait pas complet sans revenir sur un point évoqué plus haut dans ce document : le découplage entre interface et implémentation. Sur un plan technique, au niveau le plus bas, il est rendu effectif par l'utilisation des standards d'interopérabilité SOAP et WSDL. Cette dichotomie doit se retrouver à tous les niveaux, dans tous les assemblages, jusqu'au processus métier.

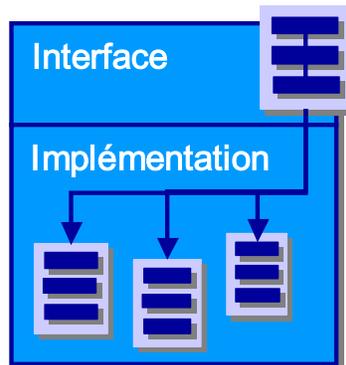


Figure 10 – Séparation entre interface et implémentation

Citons le cas d'une entreprise internationale qui souhaite normaliser certains processus, les processus d'achat par exemple. Il sera difficile pour le groupe, voire irréaliste, de contraindre chacune des filiales à respecter exactement le processus défini : la façon de faire de chacune des filiales a sa raison d'être et ne peut être uniformément rationalisé sans soulever des questions d'efficacité ou d'interventionnisme. On se retrouve ainsi presque dans une configuration B2B, avec un processus public partagé et un processus privé propre à chacun des acteurs ; sauf qu'ici, contrairement au monde B2B, on n'est pas en présence de processus privés

qui, séquentiellement alignés, décrivent un processus public complet, mais plutôt en présence de versions spécifiques pour chaque acteur d'un même processus, qui doit prévoir des points de coordination avec le processus public. Alors, la façon dont le processus s'exécute (son implémentation) dans chacune des filiales est spécifique même si, pour le groupe, l'interface du processus est unique.

De niveau fonctionnel le plus haut bas au niveau technique de granularité la plus faible, on peut reproduire le modèle de découplage entre interface et implémentation, où l'interface constitue une couche d'abstraction, soit fonctionnelle, soit technique, en fonction du modèle défini.

Si cette dichotomie entre interface et implémentation peut paraître naturelle et évidente sur le plan technique. Il est ainsi intéressant de noter qu'elle s'accompagne d'une réflexion similaire sur le plan fonctionnel. Le modèle d'architecture ci-dessous représente la dualité des couches d'abstraction.

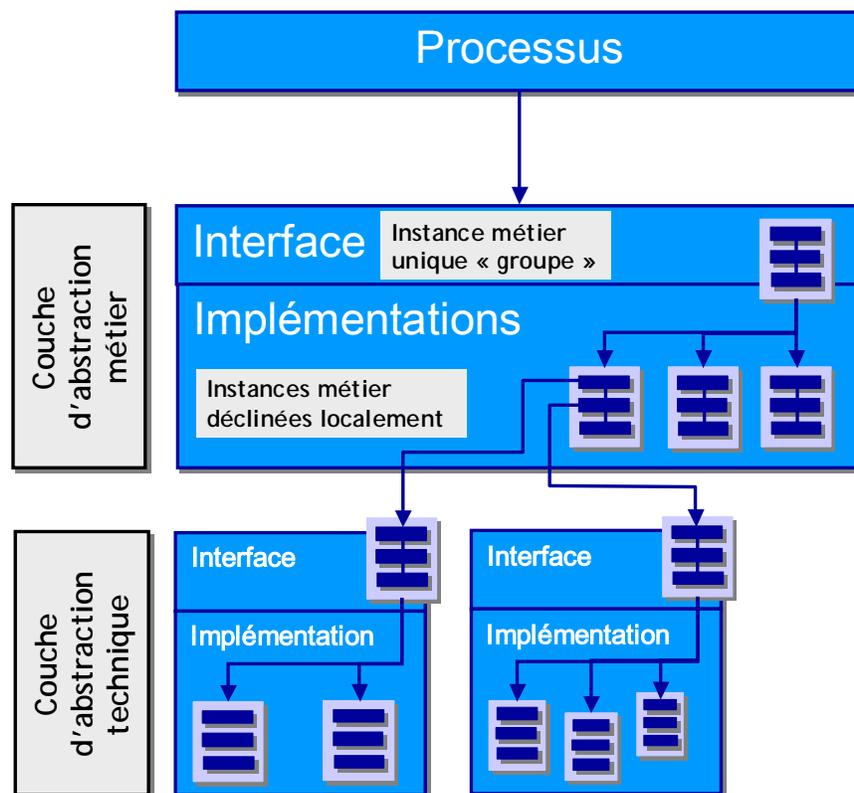


Figure 11 – La dualité des couches d'abstraction

La couche d'abstraction métier cherche à représenter les processus sous leur angle générique, telle que l'entreprise dans son ensemble souhaite les voir. Pour autant, il est certain que pour une interface de processus, l'implémentation sur le terrain va différer : par exemple, deux filiales d'un même groupe vont probablement travailler différemment. L'objectif de la segmentation entre interface et implémentation est de tenir compte des spécificités locales, tout en adoptant une vision groupe qui permettra la consolidation des informations.

La logique est identique concernant la couche d'abstraction technique : deux systèmes d'informations de deux filiales différentes vont probablement reposer sur des applications et des

technologies différentes. Malgré tout, la représentation des services doit être identique sur le plan de leur interface.

Cette architecture a été mise en œuvre sur le produit d'Intalio chez certains *early adapters* français. Les enjeux sont plutôt à chercher du côté de l'alignement métier et de l'urbanisme, ce qui se traduit par une volonté de standardiser les langages liés à la description et à la modélisation des processus métier.

2.2 Une construction orientée Processus

Métier ou IT, les processus forment l'un des axes prioritaires de développement des standards et des SOA. L'objectif recherché est l'orchestration et l'assemblage de services, pour définir des entités fonctionnelles génériques de haut niveau, qui tireront le système d'information vers le haut et privilégieront l'alignement métier, par accroissement des capacités de supervision du fonctionnement du système d'information et la performance de ces processus.

De plus en plus d'entreprises en France sont convaincues de cette nécessité. Cette volonté pousse à promouvoir des standards qui assureront un lien entre modélisation métier et exécution opérationnelle, et un portage bidirectionnel des processus définis, quelle que soit la plate-forme technologique sous-jacente.

Destinés à améliorer la portabilité des processus entre applications (par exemple, permettre le transfert d'un processus entre un atelier de modélisation et un moteur d'exécution), mais aussi à permettre à des partenaires B2B de partager la description de séquences métier, les principaux langages sont au nombre de trois :

- le premier, **BPMN** (*Business Process Modelling Notation*), concerne la normalisation de la représentation graphique des processus depuis tout atelier graphique ;
- le second, **BPEL** (*Business Process Execution Language*), décrit la structure des documents XML où figurent, entre autres informations, les étapes métier du processus, leur séquence d'enchaînement et les acteurs impliqués,
- et enfin le troisième, **BPQL** (*Business Process Query Language*), permet l'interrogation et l'extraction de fragments d'information depuis tout document BPEL, pour connaître notamment le statut et l'état d'avancement d'un processus.

Notation	Execution	Query
BPMN	BPEL	BPQL

La notation avec BPMN

Jusqu'à il y a peu, dans le petit monde de la représentation des processus, chacun, éditeur de logiciel ou cabinet de conseil, était libre de choisir ou définir le formalisme utilisé pour faire passer au mieux les messages entre experts métier, responsables fonctionnels, équipes techniques... Après tout, un formalisme est avant tout employé pour structurer la communication entre les acteurs, et dès lors que toutes les parties concernées s'accordent sur un mode de représentation, on peut considérer que l'objectif est atteint. Et voilà comment est apparu, dans chaque produit de modélisation (ou presque), un formalisme propre à chacun des éditeurs, sans

qu'on puisse, finalement, y trouver grand chose à redire. En effet, tant que les besoins d'échanger les éléments de ces référentiels de cartographie étaient faibles, l'absence de normalisation et de standards ne pouvait se révéler préjudiciable.

Mais, avec le jeu des fusions/acquisitions, avec l'essor du B2B et des formalisations de processus publics entre partenaires, avec le besoin de faire communiquer entre eux des outils aux périmètres complémentaires (modélisation, exécution, exploitation) pour partager les informations de leurs référentiels respectifs et dérouler le cycle de vie du processus métier, est apparue la nécessité de standardiser non seulement les modes d'exécution des processus, mais également leur notation.

C'est l'objectif de la norme BPMN (Business Process Modelling Notation), norme de modélisation graphique de processus métier, promue par l'organisme BPML.org. Elle repose sur la création de BPD (Business Process Diagram), dans lequel figurent plusieurs types de composants parmi lesquels :

- Les flow objects : ce sont les éléments de base des BSD. Les activités (*activity*) forment l'élément principal, sous-forme de tâche unitaire (*task*) ou de sous-processus (*sub-process*). Cette notion de sous-processus amène avec elle les possibilités d'assemblage de services et de chorégraphies. Enfin, les *gateways* correspondent à des nœuds de prise de décision, des temporisations, des *forks*, des *joins*...
- Les connecting objects : ce sont les éléments qui effectuent les liaisons entre les *flow objects*.
- Les swimlanes : les « lignes de natation » qui représentent chacun des acteurs du processus.
- Les artifacts : permettent de commenter les éléments précédents afin de préciser leur rôle dans un contexte donné.

BPMN semble d'ores et déjà promis à un bel avenir et affichent quelques points forts significatifs :

- Complétude de la modélisation ;
- Compréhension du formalisme par les différents acteurs amenés à travailler autour des BSD, de l'analyste métier au développeur ;
- Un intéressant mapping avec BPEL4WS (Business Process Execution Language For Web Services – voir ci-dessous). La spécification BPMN fournit un tableau de correspondance qui indique, au cas par cas, la traduction de chaque élément BPMN en langage BPEL.

Nombreuses sont les sociétés ayant conformé ou annoncé conformer leurs produits à cette notation. Parmi ceux ayant déjà effectué le portage, citons entre autres : aXway Process Manager™, ILOG JViews™, Popkin's System Architect™, et SeeBeyond's Integrated Composite Application Network (ICAN) Suite™. Microsoft Visio 2003™ fournit également un template pour BPMN 0.9.

D'autres annoncent un portage prochain, et parmi ceux-ci on retrouve les produits Casewise Corporate Modeler™, IBM's WebSphere Business Integration Modeler™, IDS-Scheer's Aris™, Intalio n³ Designer™, Mega Suite™, et Staffware's Process Suite™.

L'exécution des processus avec BPEL

Le travail conjoint de Microsoft, IBM et BEA dans le domaine de la description de l'exécution des processus métier a conduit à l'apparition de BPEL4WS (Business Process Execution Language For Web Services), acronyme à rallonge apparu courant 2002, devenu depuis BPEL, dans le sillage de l'effervescence éruptive autour des Web Services. Depuis avril 2003, la normalisation du langage est passée sous la responsabilité du comité technique WS-BPEL de l'OASIS. Ce groupe de travail comprend aujourd'hui des experts des principaux acteurs mondiaux de l'édition logicielle ERP et EAI, parmi lesquels on peut citer BEA, IBM, Microsoft, SAP, SeeBeyond ou encore Siebel.

En plus de la description des activités d'un processus, l'objectif de ce langage est de rendre compte de l'état d'exécution (statut) et d'avancement des processus. Ce langage est donc exploitable non seulement au niveau du modèle, comme BPMN dont il peut hériter les descriptions, mais également au niveau des instances. Cependant, malgré le *mapping* entre BPMN et BPEL, la notion de processus est légèrement différente entre les deux normes. Elle est davantage structurée pour BPMN, à cause de la hiérarchie qu'offre la possibilité de représenter des sous-processus. D'autre part, un processus exécutable BPEL différent est associé à chacune des swimlanes d'un processus BPMN.

Auparavant, quelques initiatives « locales » étaient venues signifier les prémices de cette normalisation. Dans un premier temps, Microsoft s'était fendu, avec XLANG, d'un langage XML destiné notamment à assurer la passerelle entre la modélisation des processus dans Visio et leur implémentation dans BizTalk. Ainsi, le processus n'était défini et décrit qu'une fois pour toutes, dans un atelier unique, par des experts métier, pour être ensuite exporté vers sa plate-forme d'implémentation manipulée par des experts du développement. A chaque public son outil : c'est toute la logique de Microsoft que l'on retrouve derrière ce mode opératoire. On le retrouve notamment dans les modes opératoires liés à la mise en œuvre des outils de BAM (Business Activity Monitoring). IBM avait alors emboîté à l'éditeur de Redmond en promouvant WSFL (Web Services Flow Language). WSFL semblait ainsi focalisé, jusque dans son nom même, sur les Services Web, alors que la vocation de XLANG semblait plus large.

Les deux géants se rapprochaient pour signer alors, avec le concours de BEA, troisième acteur incontournable, ce qui allait devenir BPEL4WS, initiative qui se voulait la norme unique dans la description des processus métier d'entreprise. A priori, on ne pourrait que louer une telle volonté de standardisation, si ce n'est qu'elle s'est opérée au détriment de BPML (Business Process Modelling Language), initiative nettement plus complète et stabilisée, qui n'a, elle, qu'un léger défaut : celui d'être promu par des acteurs de taille nettement plus modeste que celle des trois grands éditeurs cités (et principalement un petit nouveau au destin prometteur : Intalio, que nous évoquerons plus en détail dans la suite de ce document). L'intérêt et l'avantage de ces standards sont de demeurer encore à ce jour d'utilisation complètement libre de droits. Le principal avantage de BPML, son antériorité, en faisait un standard plus complet que ce que pouvait proposer BPEL4WS dans sa première version. Il devenait inévitable que les initiatives fusionnent pour enrichir BPEL4WS, désormais standardisé dans sa version 1.2 sous le nom BPEL ou WS-BPEL.

Aujourd'hui, BPEL continue de se stabiliser, pour continuer d'intégrer dans sa spécification les travaux en cours sur le socle technique des Web Services, concernant des aspects transversaux

comme le messaging, la sécurité ou les transactions. Tant que les langages afférents sont en cours de définition, il est inévitable que BPEL ne puisse être finalisé. Ainsi, si la spécification du langage est soumise à standardisation auprès de l'OASIS, les aspects transactionnels (WS-Coordination), pourtant fondamentaux, n'ont pas encore suivi le même chemin. On trouvera ainsi généralement dans les moteurs d'exécution BPEL (dénommés aussi dans la suite de ce document Orchestrateurs), une implémentation native des fonctionnalités de compensation et de gestion des exceptions.

Ainsi, les limitations du langage ont conduit, comme souvent, à des implémentations propriétaires dans les moteurs d'exécution, ou à des contournements comme celui proposé conjointement par BEA et IBM, avec le langage BPELJ, soit l'intégration dans le corps d'un document BPEL de blocs de traitements en Java : les *snippets*. Ceux-ci résolvent des limitations comme la gestion des variables, mais à quel prix ? Doit-on s'attendre à trouver des déclinaisons similaires pour d'autres langages, tels que PHP, ou pour le framework .NET ? On peut le penser, mais on peut aussi le déplorer, car la portabilité d'un processus BPEL s'en trouverait impactée.

Le requêtage avec BPQL

BPQL (Business Process Query Language) est un langage de requête qui a pour vocation l'interrogation de documents XML dédiés aux processus et la supervision des processus. Par interrogation, nous entendons l'extraction de fragments de documents XML (sélections, jointures...) ou de valeurs d'informations.

Il est issu des travaux liés à un autre langage d'interrogation (Stack Based Query Language) et, plus généralement, d'une agrégation des langages d'interrogation XML traditionnellement utilisés (XPath, Xquery...). Ce langage s'appuie sur un méta-modèle objet, extension du méta-modèle standard du WfMC (Workflow Management Coalition).

BPQL a pour vocation de devenir le moyen privilégié par lequel on peut connaître à tout instant l'état d'avancement ou le résultat de l'exécution d'une instance de processus. Malheureusement, il semble qu'à ce jour aucun lien clair ne soit établi entre BPQL et BPEL, qui devrait pourtant devenir son terrain de prédilection.

Il apparaît ainsi que ce langage peut être appelé à jouer un rôle important dans la supervision de l'activité métier (BAM), en temps réel ou non.

3. Apports des SOA à l'alignement métier et l'urbanisme

Nous allons dans cette partie étudier les apports des Architectures Orientées Services à l'alignement métier des systèmes d'information et, par conséquent, à des notions proches de ce que l'on nomme l'urbanisme ou l'urbanisation des systèmes fonctionnels.

Pour faire court, rappelons que ce terme désigne un ensemble de méthodes et de techniques dédiées à la réorganisation fonctionnelle des systèmes d'information, dans le but, notamment, de les rationaliser, de les homogénéiser, afin d'aboutir à une plus grande liberté d'action sur le pilotage des données et des processus, et, finalement, à une flexibilité accrue et à un alignement métier facilité. Nous vous invitons à parcourir la littérature parue sur ce thème pour de plus amples informations.

3.1 Les outils de la rationalisation et de la réutilisabilité

À la lumière des cas d'urbanisme évoqués ici, il devient clair qu'au delà d'un système piloté par un orchestrateur adapté, il est indispensable d'utiliser pour la conception des SOA un outillage amont adéquat, autant pour modéliser et cartographier métier et technique que pour assurer, via des passerelles d'échange, une cohérence continue entre spécification et implémentation.

Un des buts et idéaux d'une démarche d'urbanisation consiste à rationaliser les éléments fonctionnels dont le périmètre est identique, et à éviter leur multiplication, afin de supprimer toute redondance aux quatre coins du système d'information, et disposer ainsi d'un SI homogénéisé, pilotable, plus simple et moins coûteux. C'est également l'une des volontés qui doit présider à la construction d'une SOA. La possibilité de définir des services d'entreprise et des services transversaux semble aller dans cette direction, impression renforcée par la mise à disposition, autour des Services Web, d'un annuaire central d'entreprise au format UDDI.

Pour parvenir à cet objectif, il convient de cartographier efficacement les différents services fournis dans l'entreprise. L'organisation d'une plate-forme de services et la recherche de réutilisabilité demandent de prendre de la distance par rapport au système, donc à modéliser la cible avant de la mettre en œuvre. Cette modélisation ne doit pas être perçue comme un travail détaché des réalités de la mise en œuvre : on privilégiera un outillage capable de s'interfacer avec le terrain (les serveurs opérationnels d'exécution) et de se synchroniser avec eux sur la base des standards adéquats, notamment ceux évoqués plus haut dans ce document (BPEL, XML-Schemas, SOAP, WSDL, UDDI...).

Cette cartographie vise à recenser l'ensemble des services constitutifs du socle SOA. Dans le cadre de l'établissement d'une architecture multi-dimensionnelle et multi-niveaux, la cartographie vise également à catégoriser les services pour mieux en qualifier le périmètre, la portée et l'utilisation :

- certains services sont fonctionnels et liés à des applications et d'autres sont techniques.
- certains services n'ont qu'une portée locale, applicable à un bloc fonctionnel identifié.
- d'autres sont transversaux et applicables à l'ensemble de la plate-forme.

Il existe autant d'axes d'analyse qu'il existe de méta-données pour qualifier un service dans l'annuaire qui les référence :

- Le type ou la classe du service : service de présentation, d'accès aux données...
- Le sujet adressé : applications, données, flux, réseau, systèmes, postes de travail...
- Le bénéficiaire du service : ressource machine et/ou personne physique, avec les droits d'accès correspondants,
- Le type d'usage,
- Les technologies utilisées couvrant le service, notamment dans le cadre de la constitution d'un référentiel d'entreprise.

Cette liste n'est pas exhaustive, et selon les différentes vues souhaitées, l'outil de cartographie établira les différentes matrices de liaisons croisées et d'analyse d'impact.

On sait qu'un travail de recensement et de cartographie des services en amont est un pré-requis indispensable au maintien de l'évolutivité de la plate-forme et aux décisions d'alignement métier, ainsi qu'aux rationalisations et réutilisations nécessaires à l'obtention du retour sur investissement. Cependant, une fois ce travail effectué, il ne doit pas devenir obsolète : les modifications de modélisation doivent être rapidement intégrées, et les changements au niveau du terrain doivent être reportés, après validation, au niveau de la cartographie.

Cet outil sera donc équipé d'un véritable référentiel avec des passerelles de synchronisation au niveau du terrain pour assurer une mise à jour bidirectionnelle, sur la base des standards de l'industrie, établis ou en passe de l'être (XML-Schemas, BPMN (Business Process Modelling Notation), BPEL (Business Process Execution Language)...). La présence d'un référentiel dynamique et interrogeable doit être recherchée. Ainsi les outils de modélisation plats, sans référentiel (Powerpoint, Visio...) devraient ainsi être évités, au profit de produits tels que Aris d'IDS Scheer, Isiman de Keyword, Corporate Modeler de Mega, System Architect de Popkin Software, et bien d'autres encore. Ces produits ne sont pas tous à l'heure actuelle fournis avec la capacité de modéliser et stocker une bibliothèque de services ; mais tous ont un méta-modèle personnalisable qui peut s'adapter à celui défini par une entreprise pour la représentation et la catégorisation de ses services.

On comprend mieux dans quelle mesure une démarche de construction de SOA s'inscrit en plein dans le volonté généralisée de rationalisation et de gouvernance que recherchent les entreprises. La réutilisabilité des services est plus concrète que n'a jamais pu l'être celle des composants. La construction d'une SOA à un niveau d'entreprise demande de l'organisation et de la méthode, relayés par des outils adaptés, pour :

- **cartographier** l'utilisation des services, les liens existants entre les services invoqués, les versions des services et des processus utilisés, dresser des analyses d'impact...
- **orchestrer** l'exécution des processus et des chorégraphies, tracer le résultat de cette exécution, notifier des erreurs et anomalies rencontrées, délivrer un *reporting* régulier en mesurant la qualité de service...

Sans ce travail parallèle, il est évident que la multiplication des services et des niveaux de services entraînera une absence de visibilité tant au niveau de la conception, de l'exécution que de l'exploitation, qui serait préjudiciable au système d'information tout entier et serait vécu comme un véritable retour en arrière.

Or, signe des temps, la demande est actuellement très forte en ce qui concerne l'outillage pragmatique d'une démarche d'urbanisation vers un champ d'application à mi-chemin entre organisation de SOA et intégration d'applications. Qu'en déduire ? Que le lien entre modélisation/cartographie et orchestration est plus fort que jamais, relayé par l'émergence de standard et le besoin d'organiser réellement et concrètement les systèmes d'information dans le sens d'une plus grande rationalisation et d'une meilleure réutilisation des services existants.

Le découplage entre Système d'Information et Système Informatique, qu'introduit la SOA, et le besoin constant envers les outils de modélisation, tend bien à prouver que la mise en œuvre de SOA constitue un facteur facilitant de l'alignement métier.

3.2 Les apports à l'urbanisation

L'urbanisation itérative

Nous allons ici montrer comment les différents types de services et leur articulation au sein d'une SOA permettent de faire progresser les traditionnels modes de représentation des systèmes d'information. Un des schémas les plus connus est celui de la figure 12, qui met en parallèle différents niveaux d'architecture.

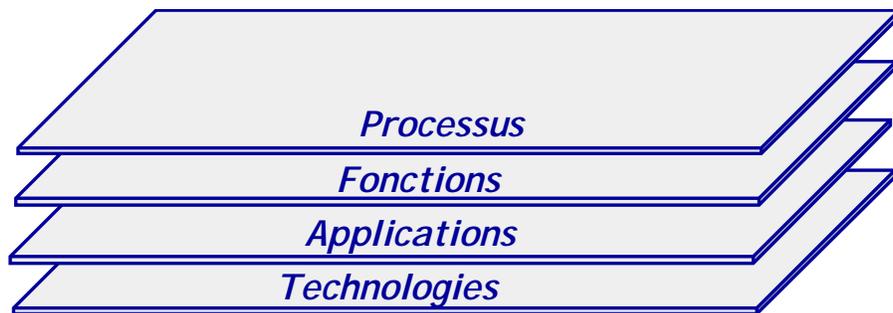


Figure 12 - Niveaux parallèles d'architecture

La logique de cette figure est de montrer que les processus sont composés de Fonctions, elles-mêmes hébergées dans des Applications, composées de Technologies. L'idée est de montrer que des liens verticaux existent, et que l'on peut, par exemple, relier un processus à un ensemble de technologies. Système d'information et système informatique sont ainsi liés dans ce mode de représentation que de nombreux travaux d'urbanisme reprennent comme un passage obligé.

Par les SOA, les applications deviennent des blocs granulaires découpés en fonctions, accessibles séparément les unes des autres. Le principal intérêt de la notion de Service est ainsi de dépasser la notion d'application, ce bloc monolithique technique, pour accéder à un grain fonctionnel plus précis. En réalité, ce mécanisme existe depuis longtemps : d'abord réservé à des fonctions techniques de système d'exploitation, l'accès aux API (*Application Programming Interface**) s'est étendu aux progiciels du marché. Le discours SOA peut ainsi être vu comme une volonté de généraliser cette démarche à tout type d'application et d'accès aux données. Les prochaines versions des grandes bases de données du marché sont ainsi attendues avec des fonctionnalités d'exposition de services et, à terme, toute source de données sera accessible via ce type d'accès (volonté déjà présente précédemment dans certaines initiatives comme OLE DB de Microsoft).

En cela, les services viennent insérer un niveau intermédiaire de représentation dans les schémas traditionnels.

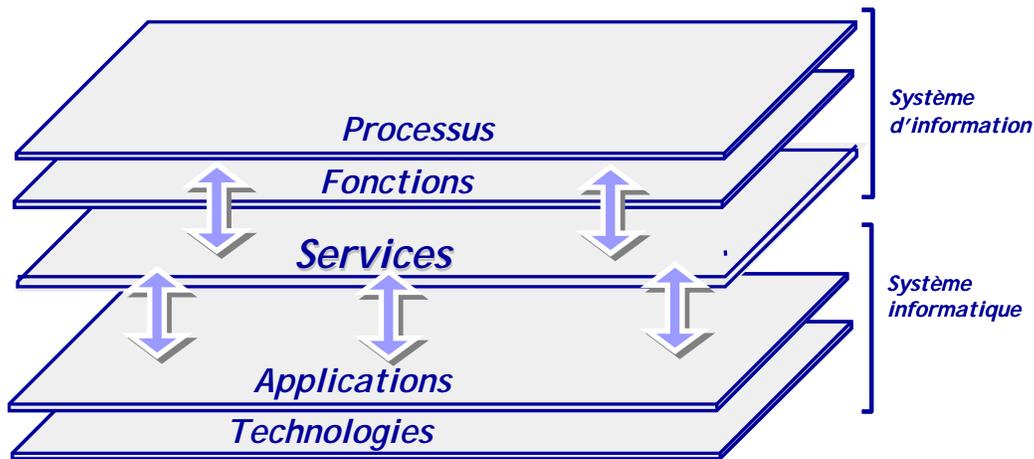


Figure 13 - Insertion de la couche de Services

Si la nature fonctionnelle du Service peut être discutée, on peut cependant s'en tenir à cette définition. Dans ce cas, on s'aperçoit que cette couche assure la transition (autrefois davantage forcée) entre la couche Fonctions et la couche Applications : les Services sont hébergés (ou non, d'ailleurs) dans des Applications. D'un côté, on se situe dans le métier et dans le Système d'Information, de l'autre dans la Technologie et le Système Informatique.

Dans l'ouvrage « L'EAI au service de l'entreprise évolutive », nous avons déjà évoqué ce besoin de découpler les niveaux fonctionnels des niveaux techniques. L'identité Fonction/Application ne pouvait être assurée en 1 à 1 et traduisait en réalité les contraintes qui pèsent sur les systèmes d'information et freinent la flexibilité des processus et des règles métier. Avec les SOA, nous disposons de principes qui concrétisent cette volonté, en établissant cette distinction recherchée entre le service et son implémentation, entre le service et son contenant : l'application.

Evidemment, tant qu'un modèle de Service n'est pas généralisé et pérennisé partout dans le Système Informatique, cette latitude n'est jamais que partielle. Mais, comme toute volonté d'urbanisation, elle est forcément progressive et tend vers une cible, que l'on souhaite atteindre dans la mesure où tout pas vers elle constitue un progrès. Le fait de l'atteindre à terme sera un achèvement, et, dans l'attente, des bénéfices seront cependant obtenus à tout pas effectué. C'est donc de façon progressive, et encore momentanément contrainte par la généralisation des implémentations de services dans le SI, que s'effectue leur urbanisation à l'aide des SOA.

Dans ce but, on recherchera, parmi les fonctionnalités attendues d'un orchestrateur SOA, la capacité à rendre accessible facilement, sous forme de services, les grains fonctionnels de systèmes n'ayant pas été conçus dans ce sens, et notamment les systèmes légaux mainframe. A titre d'exemple de produits fournissant ces fonctionnalités, citons le couple BizTalk Server/Host Integration Server de Microsoft, qui normalise technologiquement les accès MVS et AS400 iSeries sous forme d'appels SOAP.

Ces fonctionnalités permettent non seulement d'accélérer la généralisation d'une SOA, mais aussi de mieux piloter son plan d'urbanisation, sans nécessairement attendre qu'un éditeur livre une nouvelle version orientée service de ses solutions. L'orchestrateur SOA devient levier

d'urbanisation : son interfaçage avec des ateliers de modélisation et des outils de cartographie n'en devient que plus important.

La migration itérative des applications obsolètes

Le remplacement d'une application par une autre, au périmètre fonctionnellement identique, revêtait, avant l'arrivée sur le marché des EAI, un caractère assez brutal, qui se traduisait généralement par une migration *one-shot* des données et l'extinction définitive de l'application obsolète au profit de sa remplaçante.

Les technologies de l'EAI, dans leur capacité à piloter l'intégration d'applications, apportaient davantage de souplesse : une fois les données migrées, les deux applications pouvaient continuer à cohabiter, l'EAI assurant la synchronisation régulière des données, en temps utile. La migration des fonctions pouvait alors s'effectuer au rythme des besoins et des possibilités.

Pourtant, cet intéressant modèle n'a finalement été que peu utilisé. Performant, il n'apportait pas une complète flexibilité, limité qu'il était par l'imbrication persistante entre fonction et application. L'arrivée des SOA, qui distingue ces deux notions, pour mieux se concentrer sur la fonction, devrait lui redonner un second souffle.

De plus, l'arrivée en fin de vie de certains logiciels demande parfois d'opérer une activité de *reverse engineering* avant d'œuvrer à leur remplacement, avec externalisation temporaire de tout ou partie du processus au sein de l'orchestrateur de service. La modélisation de certains des processus couverts par le logiciel obsolète, et son outillage au moyen d'une plate-forme d'orchestration constitue un substitut temporaire, voire permanent, au logiciel lui-même :

- Dans un premier temps, le logiciel conserve ses données mais certaines de ses fonctionnalités sont progressivement externalisées dans la plate-forme d'orchestration.
- Dans un second temps, on opère une migration définitive des données du logiciel en fin de vie vers son remplaçant. Les processus sont alors progressivement activés par leur implémentation dans la plate-forme. La continuité de l'activité est assurée par ce travail de modélisation qui peut être ensuite factorisé vers un atelier de modélisation tiers pour adonner lieu à une forme de cartographie qui pourra évoluer vers un périmètre d'entreprise.

Ces cas de figure peuvent également se présenter dans le cas d'une fusion, lors du remplacement d'un ERP par un autre.

La Figure 14 ci-dessous illustre ce contexte.

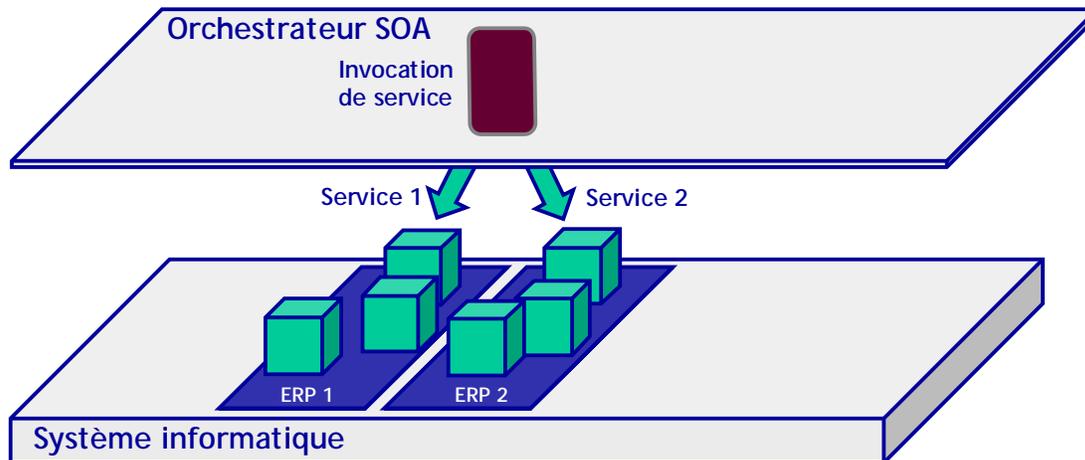


Figure 14 – Cohabitation de deux applications iso-fonctionnelles

Dans ce schéma, deux ERP exposent deux services fonctionnellement identiques, le Service 1 et le Service 2, qui insèrent ou mettent à jour des données dans leurs modèles de données respectifs. Tant que ces deux systèmes cohabitent, il est nécessaire que toute mise à jour des données de l'un s'accompagne d'une mise à jour des données de l'autre. Les mêmes données sont donc passées à chacun des deux services. Pourtant, rien ne garantit que les interfaces d'accès à ces services soient les mêmes : les éditeurs étant différents, les modèles de données le sont fatalement, et il n'existe pas aucun standard, aucune nomenclature, pour normaliser les contrats d'interface de services fonctionnellement identiques. Si les données à passer sont les mêmes, leur formatage en vue d'invoquer les services sera différent. Le processus à suivre pour maintenir la cohabitation des systèmes est le suivant :

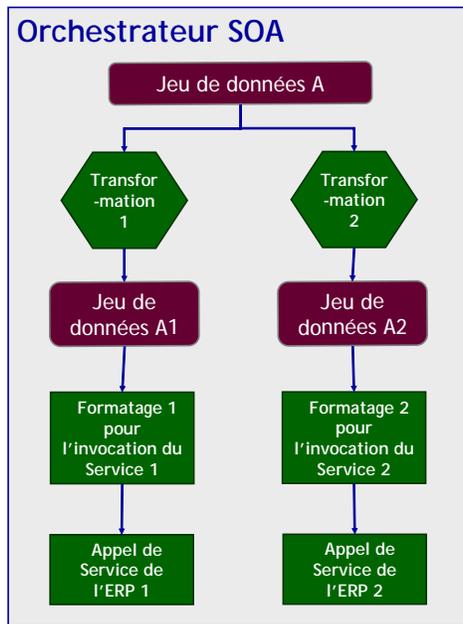


Figure 15 - Invocation parallèle des services

Cette figure n'a pour objectif que de montrer les fonctionnalités requises pour un orchestrateur SOA :

- Des fonctions de transformation et de formatage des données,
- Des fonctions d'invocation de services, donc la capacité à lancer des appels SOAP.

Ces deux types de fonctionnalités (transformation et connectivité) se retrouvent dans plusieurs catégories d'outils, qui seront abordées dans la quatrième partie de ce document.

On s'aperçoit d'autre part au travers de cet exemple, que ce travail de remplacement ou de cohabitation d'applications fonctionnellement proches :

- N'est pas gêné par la non-standardisation des contrats d'interface, dans la mesure où les orchestrateurs SOA comprennent nativement les moyens d'y pallier.
- Mais qu'il peut se heurter à l'absence de nomenclature dans la description fonctionnelle des Services.

En effet :

- Dans le meilleur des cas, on a beaucoup de chance et il y a identité fonctionnelle entre les services des deux systèmes
- Les soucis commencent dès que le périmètre fonctionnel n'est plus identique, et là, tous les cas de figure sont envisageables pour pallier à cette difficulté, mais cela impose un travail supplémentaire

Dans tous les cas, on ne sera pas exempt, en l'absence d'une nomenclature de service qui, à notre connaissance, n'est pas l'objet d'une concertation entre acteurs de l'industrie (les principaux éditeurs d'ERP, par exemple), de procéder à un travail approfondi d'étude des services respectivement exposés par les deux systèmes, afin d'en établir la correspondance certaine, et connaître les besoins afférents dans la transformation des données. Ce travail amont

peut à nouveau être réalisé depuis un atelier de cartographie de services, qui doit notamment proposer des fonctionnalités d'analyse d'impact entre situations sources et cible.

La construction de référentiels d'entreprise

L'une des grandes réflexions menées actuellement dans bon nombre d'entreprises concerne la création de référentiels d'entreprise (produits, clients, fournisseurs), vastes ensembles d'objets métier centralisant l'intégralité des informations du modèle public partagé entre tous les départements fonctionnels et techniques, et qui recentrent les préoccupations sur la gestion des données.

Il existe plusieurs façons de structurer et de mettre en œuvre de tels référentiels. Certains outils permettent déjà de collecter les données dans des sources hétérogènes éparses, de les fédérer et de les analyser pour identifier les doublons et définir une nomenclature. On trouve dans cette catégorie le produit MDM (Master Data Management) de SAP, celui d'Oracle (également nommé MDM) et l'offre Product Center d'IBM, davantage centrée, comme son nom l'indique, sur le référentiel Produit (rachat de la société Trigo). Il est intéressant de noter le positionnement d'IBM ici, dans la mesure où cet éditeur ne propose pas de progiciel intégré : cela indique clairement que la gestion des données de référence ressort autant de l'organisation et du métier que de l'infrastructure. En effet, une fois le référentiel constitué et opérationnel, il doit communiquer avec les applications du système d'information pour les approvisionner en données de référence suite à tout ajout, modification ou suppression d'information. Cette propagation d'information prend la forme de processus de synchronisation.

- Certains événements métier entraînent la création ou la mise à jour d'objets d'entreprise, potentiellement hors du référentiel : ils doivent lui être répercutés pour validation par des utilisateurs métier. Ce flux de données ne doit pas s'opérer directement, par accès direct au modèle de données, mais via un service qui prévoit ou non une action humaine sur les données, avant leur intégration.
- Le référentiel à jour, un sous-processus doit prendre le relais pour assurer la propagation des données auprès des applications clientes, soit par transmission directe de données, soit par notification de mise à disposition des d'informations. Dans le premier cas, un processus/service de synchronisation pilote cette diffusion, de préférence via un modèle événementiel.

Dans tous les cas, ces flux de données peuvent être pris en charge par des systèmes de type EAI, mais les interfaces entre le référentiel et le reste du système d'information ne se réalisent jamais par un accès direct au modèle de données. En entrée comme en sortie (au niveau des modèles de connexion), c'est un modèle de services qui est exploité.

Rien n'oblige technologiquement à gérer ce point sous l'angle du service : c'est une volonté d'urbanisation qui veille en fait à l'exploitabilité, à la maintenance, à l'évolutivité du système d'information, qui s'applique ici. Si l'accès au référentiel s'opère directement sur les données, toute modification dans la structure de la base aura des impacts sur les processus d'échange. Si on travaille au niveau des services, ceux-ci constituant des interfaces contractuelles, les modifications du modèle de données doivent garantir la compatibilité ascendante.

Dans le second cas, c'est l'application client qui va elle-même, le moment venu, invoquer un service d'appel aux données du référentiel, ce qui implique :

- Que le référentiel d'entreprise expose lui-même son propre jeu de services ;
- Qu'il sache constituer à la demande, ou stocker, les vues correspondant aux attentes et aux besoins de chaque application.

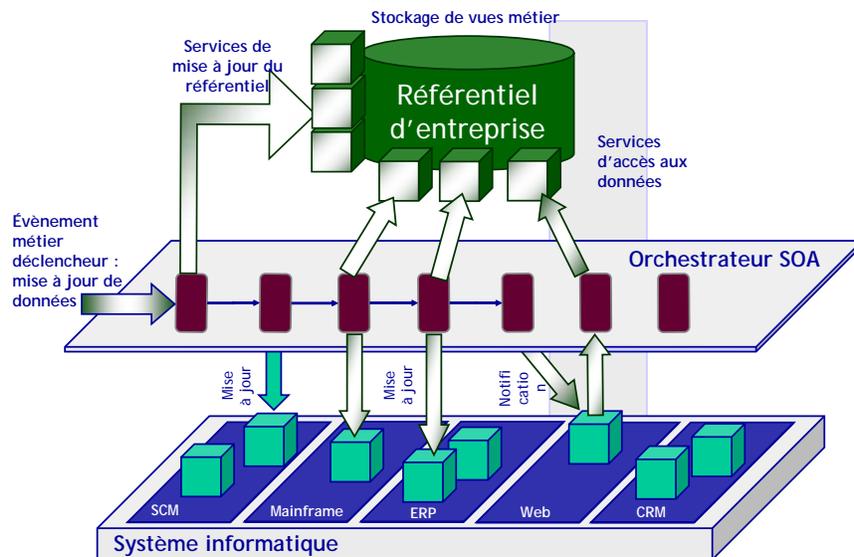


Figure 16 - Gestion SOA du référentiel d'entreprise

L'intérêt de rendre le référentiel accessible sous forme de services est de rendre celui-ci technologiquement neutre auprès des applications, et, plus important, de rendre celles-ci indépendante de la localisation du référentiel. L'annuaire de Services de l'orchestrateur SOA, et ses fonctions d'administration, n'en deviennent que plus importantes dans ce modèle, pour faire du référentiel d'entreprise créé un véritable métronome du système d'information, à des années-lumière d'une vague implémentation éloignée de toute réalité terrain.

L'assemblage d'applications composites

L'assemblage d'applications composites est un autre thème récurrent du discours sur les SOA. Il consiste à considérer qu'une application entière peut être formée d'assemblage de services, de chorégraphies et, par conséquent, de processus, logiquement regroupés par périmètre fonctionnel. Ainsi, si le système d'information tout entier bénéficie du découplage entre Service et Applications, la notion même d'application s'en trouve finalement impactée. On sort d'une situation où les blocs fonctionnels sont dépendants des contraintes liées aux applications, pour entrer dans un schéma où l'application vient se plaquer au plus près des plans d'urbanisme et des quartiers définis, puisqu'elle se construit en fonction de ceux-ci.

La Figure 17 donne un exemple de création d'applications composites.

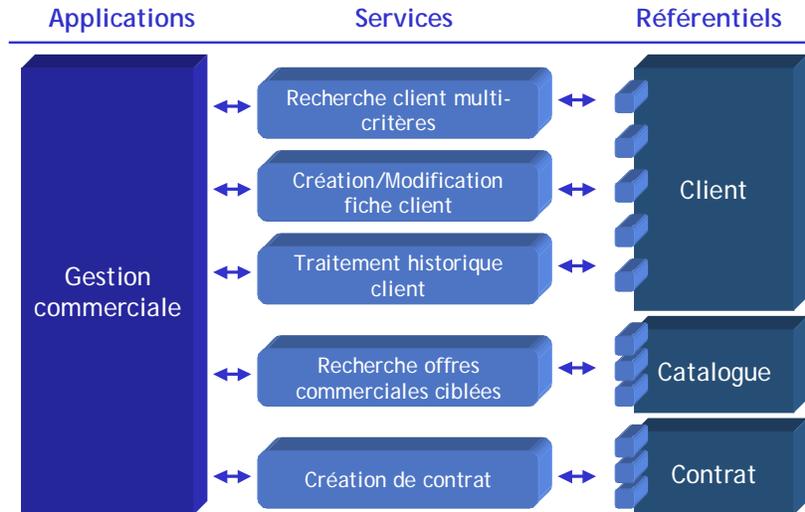


Figure 17 – Modèle de répartition Services/Données

En plus de l'application et des services invoqués, cette figure fait apparaître, à droite, les référentiels de données, ainsi que les services techniques unitaires d'accès aux données de ces référentiels. Cette représentation s'inscrit ainsi dans la logique des aspects liés aux référentiels de données évoqués ci-dessus. Elle permet de bien segmenter les rôles et les responsabilités :

- A droite les référentiels de données et leurs services unitaires d'accès.
- Au centre, la couche de services métier. En plus d'héberger leurs règles métier propres, de tels services peuvent être construits par l'invocation séquentielle de plusieurs services techniques ou d'une chorégraphie pré-définie.
- A gauche, l'application composite qui assemble les invocations de différents services métier pour construire un ou plusieurs processus métier fonctionnellement liés pour un utilisateur ou un groupe d'utilisateurs, et mis à leur disposition via une interface graphique dédiée.

Ce modèle n'est, finalement, pas bien différent du modèle 3-tier ayant accompagné l'essor des serveurs d'applications. On y retrouve en tout cas la séparation entre données, règles métier et présentation ; mais quelques années ont passé, de nouveaux concepts se sont installés, d'autres se sont affinés, de nouvelles technologies ont prévalu, et la situation est désormais la suivante :

- Les leçons de l'urbanisme et la volonté d'adresser un périmètre d'entreprise a fait de la couche de données une couche de référentiels de données ;
- L'arrivée des Web Services et des SOA a transformé la couche de composants métier en niveaux de services ;
- La progression des standards et technologies de l'Internet et la volonté d'adresser un périmètre d'entreprise ont transformé la couche de présentation en notion d'application composite.

En ce sens, la notion d'application composite semble assez proche de la notion de portail, encore qu'il faille se préserver des amalgames. Le portail, et les technologies sur lesquelles il repose, forme l'implémentation candidate la plus adaptée aujourd'hui pour implémenter ces applications.

La notion de plate-forme d'infrastructure, que nous aborderons dans la partie suivante, s'en accommode assez bien, puisqu'elle fournit des moyens mutualisé pour bâtir des portails, des Web Services métier et techniques et de multiples niveaux d'intégration et d'accès aux référentiels de données. En un sens, la notion de portail est aux applications composites ce que sont les Services Web aux SOA à l'heure actuelle : une forme d'implémentation de référence bâtie sur des standards en cours de consolidation.

Dans les sociétés de services et cabinets de conseil, nous eûmes plus d'une fois entre les mains des cahiers des charges de portail séduisants, ambitieux, mais malheureusement irréalistes compte tenu de l'état d'avancement des méthodes et des techniques. On peut estimer qu'aujourd'hui une nouvelle étape est franchie, même si l'ensemble du modèle reste encore à consolider, pour tendre à respecter les prévisions qu'émettent les analystes à la fin des années 90.

4. Eléments de méthode

4.1 Catégorisation de services

Du service dans toute chose

Où se situe le Service dans la hiérarchie des opérations d'une entreprise ? Est-il métier ? Est-il technique ? Le service n'est-il rien d'autre que l'accès à une fonction applicative unitaire auparavant insaisissable indépendamment du contexte de l'application ?

Il est courant d'évoquer une identité entre la notion de Service et la notion de Fonction. Si cette définition donne déjà quelques indications quant au niveau de granularité recherché, encore faut-il savoir ce que l'on considère comme une fonction : est-ce une fonction métier ou une fonction applicative ? Et là, on se heurte à une difficulté traditionnelle dès que l'on commence à manipuler les termes de fonction, de processus, de service. Quel en est le réel périmètre ? Qu'essaie-t-on au juste de décrire ?

En effet, une fonction d'entreprise peut être de très haut niveau et désigner une catégorie de processus métier. La fonction applicative, elle, au contraire, est par nature, localisée au sein d'une application. De même, le processus lui-même acquiert des dimensions très hétérogènes, parfois essentiellement technique, parfois métier. Enfin, la fonction, le processus et le service ne sont pas les seules notions auxquelles on se confronte dès lors qu'il s'agit de s'accorder sur un périmètre commun pour désigner ces objets. Ainsi, la Figure 18 ci-dessous décrit un exemple possible d'imbrication de différents objets manipulés, de la fonction au service, telle qu'elle est actuellement modélisée et éprouvée chez l'un de nos clients (ce découpage est par ailleurs repris dans certains logiciels de modélisation du marché).

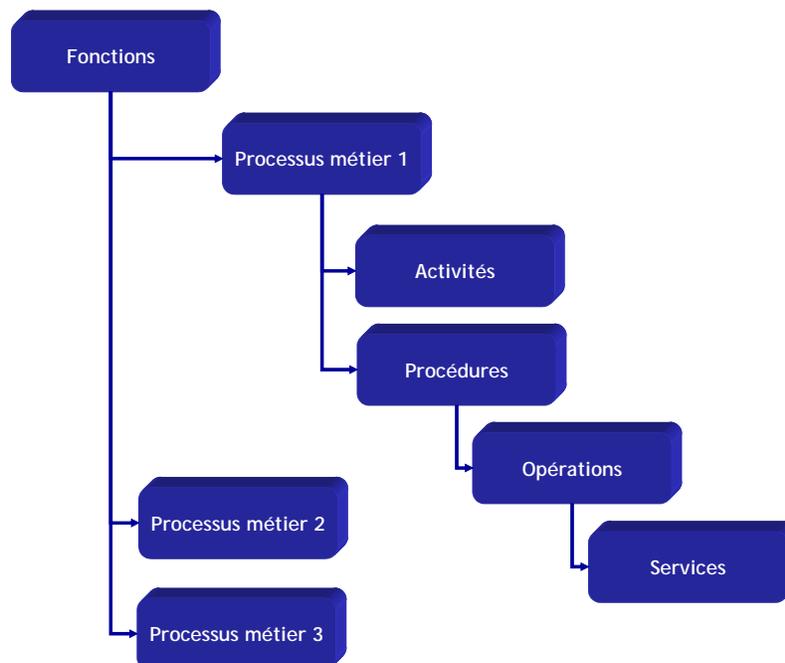


Figure 18 - De la fonction métier au service technique

Dans ce modèle, la Fonction est décrite par un ensemble de Processus Métier, conformément aux points soulevés plus haut. Ces processus combinent des Procédures, elles-mêmes représentatives d'un enchaînement d'Opérations, elles-mêmes composées d'Activités dont certaines sont associées à des services, fatalement unitaires et techniques à un tel niveau de détail. Dans cette répartition, le Service apparaît en dernier lieu, lié directement à une activité, pragmatiquement assurée par un profil d'acteur unique, et apparaît ainsi étroitement lié à l'application. Cette définition du service est fatalement réductrice.

Le service peut intervenir à tout niveau et tout élément doit pouvoir être exposé en tant que service aux acteurs qui en ont la nécessité. Il s'agit d'une hypothèse et non d'un dogme : il faut rester pragmatique et savoir d'adapter à la réalité du terrain, et surtout, aux besoins d'une entreprise. On ne peut donc pas dérouler une démarche figée et universelle : celle-ci dépendra du périmètre que l'on souhaite voir couvrir au SOA.

La contribution des SOA et des services à ces principes est de chercher à les simplifier et à les ouvrir. Chacun des éléments de ce découpage hiérarchique vont pouvoir, en fonction de ce que souhaite construire l'entreprise, être représentés sous forme de services, sur la base de standards. L'entreprise sera libre de définir le nombre de hiérarchies qui lui semble approprié, ceux qui sont modélisables ou non sous forme de services, et le nombre de hiérarchies à l'intérieur du niveau de service. Ainsi, un service peut être soit fonctionnel, soit technique, voire les deux. Un service peut être totalement encapsulé dans une application ou interagir entre plusieurs blocs logiciels. Un service peut être unitaire, constituer un assemblage de services, ou décrire une séquence d'étapes qui définissent un processus technique ou un processus métier. L'idée phare reste de privilégier un couplage lâche entre tous les niveaux de la hiérarchie, basé sur un modèle matriciel et non hiérarchique, pour briser les silos applicatifs et promouvoir la réutilisabilité recherchée.

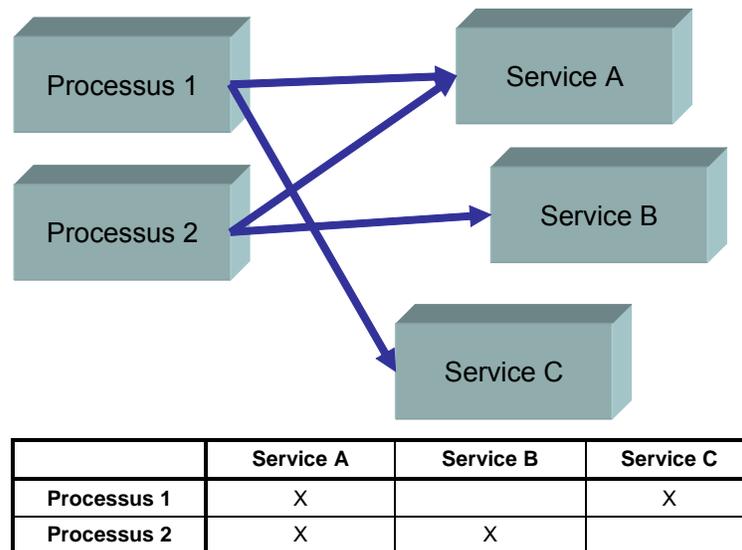


Figure 19 - Le modèle matriciel des SOA

En somme, finalement, tout est potentiellement service, ce qui est à la fois une bonne et une mauvaise nouvelle : une bonne, car on peut ainsi disposer d'un modèle généralisable et applicable de façon quasi universelle ; une mauvaise, car la multiplication d'éléments dont le

périmètre peut se recouvrir risque d'entraîner rapidement un chaos maximal. La construction réussie d'une SOA va d'abord devoir être; comme souvent, une affaire de rigueur, de discipline et d'organisation.

Ainsi, pour chaque action impactant la structure interne d'un service ou d'un assemblage, il sera nécessaire d'identifier :

- **les impacts verticaux** : imbrication hiérarchique entre niveaux de services,
- **les impacts horizontaux** : dans une plate-forme de services, certains services, comme la sécurité, sont transversaux et utilisés par d'autres services, quel que soit leur niveau dans la hiérarchie.

En raison des attentes et des niveaux d'exigence, encore plus forts que par le passé, et parce que l'alignement métier est un sujet complexe, les contraintes sur l'organisation du système d'information sont encore plus fortes que précédemment.

Verticalité de services liés aux fonctions et processus

Partant de ce postulat, on peut tenter de se livrer à un travail de catégorisation des services qui se montrera de toute façon nécessaire dans l'établissement d'une cartographie. A partir du service applicatif métier simple, ou du service technique unitaire, on peut procéder à des assemblages de services et décrire des entités techniques ou fonctionnelles de plus haut niveau :

- **Le processus automatisé** : il assemble différents services pour décrire un processus complètement automatisé et transversal aux applicatifs, qui décrivent le comportement des systèmes. Un processus de synchronisation de référentiels est un bon exemple de processus technique. Ce niveau correspond aux Opérations de la figure 18. Le langage BPEL (voir 2.2 Une construction orientée Processus) est un exemple des standards liés à l'exécution de ces processus.
- **La chorégraphie** : il s'agit d'un assemblage de service selon une séquence prédéfinie. Entité logique, la chorégraphie peut être intéressante pour décrire une transaction ou des sous-processus. Dans un contexte B2B, on y formalisera l'interaction des différents acteurs, qui prend alors la forme d'une intégration de services, donc d'assemblage de processus automatisés. Mais cette séquence de services peut également décrire l'enchaînement de services au sein d'une application unique : il s'agit dans ce cas d'une externalisation de processus, comprenant éventuellement une intervention humaine prévue dans l'orchestration. Les chorégraphies peuvent être localisées au sein d'une application unique (externalisation mais peuvent aussi, plus rarement, assembler des services d'applications séparées. Ce niveau peut correspondre aux Procédures de la figure 18.
- **Le processus métier** : c'est l'assemblage de services et/ou de chorégraphies pour décrire des séquences métier fonctionnellement cohérentes. Il peut lui-même être composé de sous-processus. Par exemple, la création d'un nouveau client au niveau de l'entreprise va consister en une étape métier (intervention utilisateur invocation de service, utilisation d'une chorégraphie) effectuée dans une application, puis il faudra synchroniser le référentiel d'entreprise, et ensuite l'ensemble des applications intéressées par cet évènement. Le langage BPMN, standard de modélisation, peut être

utilisé pour représenter ces processus. Ce niveau peut correspondre aux Activités et aux Processus de la figure 18.

Ainsi, si l'on reprend la notion de fonction telle qu'elle apparaît au début de cette partie, on s'aperçoit qu'elle est fondée sur le regroupement logique de processus métiers fonctionnellement liés. Une fonction d'entreprise devient une bibliothèque de services autant que de processus.

L'apparition du processus métier dans cette hiérarchie peut sembler inadéquate, voire tout à fait abusive. L'idée que l'on peut se faire d'un modèle de services est en effet directement lié, dans notre définition, à un système où un orchestrateur technique s'occupe d'agencer et de séquencer les activités en réponse à un ou plusieurs événements. Dans ce contexte, le processus métier connaît un cycle de vie plus ouvert, où certaines activités sont assurées par des utilisateurs humains, en dehors de toute automatisation orchestrée par le système.

Notre présentation cherche ainsi davantage à mettre l'accent sur des systèmes de BPM ou le processus modélisé dans l'orchestrateur serait davantage un processus passif en attente de notifications externes permettant d'en mesurer l'état d'avancement via des synchronisations événementielles. Nous nous trouvons ainsi encore dans un environnement automatisé, mais non moteur. Pour suivre le processus métier et disposer d'un véritable environnement de BPM, Il faut donc construire l'infrastructure de notification et les systèmes de supervision. C'est ce que proposent aujourd'hui les environnements de BAM (*Business Activity Monitoring*) et ce qui permet d'élargir le modèle d'orchestration des Services Web à un niveau de supervision métier et technique, comme l'indique la figure ci-dessous.

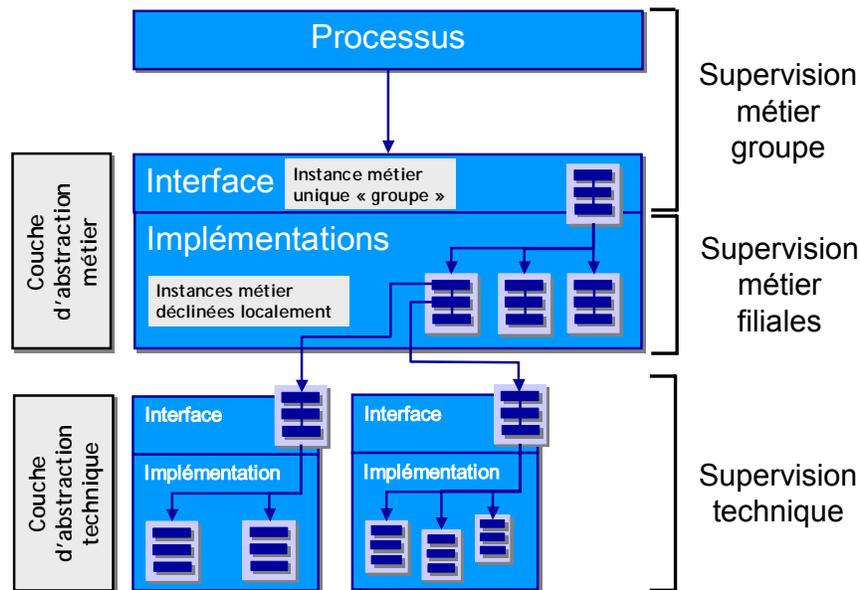


Figure 20 - SOA et modèle de pilotage et de supervision

La supervision à tous les niveaux permet de mettre en relation les événements techniques et métier, de mesurer l'impact d'un événement sur l'ensemble de la chaîne de services et de processus, voire, comme une nouvelle catégorie tend actuellement à le promettre, à effectuer de l'analyse prédictive.

Horizontalité et services techniques transversaux

La construction d'une plate-forme de services passe par l'utilisation d'un ensemble de briques de base généralement techniques, accessibles, de façon transversale, à l'ensemble des autres services. On peut ici lister, de façon non exhaustive, quelques uns de ces types de services :

- Services d'accès aux données (des référentiels, notamment),
- Services de sécurité : authentification, cryptage, journalisation, gestion des incidents...
- Services de stockage, de sauvegarde, d'archivage,
- Services d'intégration, de réplication et de consolidation de données,
- Services de transport et de routage,
- Services de transformation,
- Services d'émulation,
- Services de système d'exploitation,
- Services d'exploitation technique,
- Services de remontée d'indicateurs de performance et de pilotage fonctionnel.

On s'aperçoit qu'il est également possible d'assembler certains de ces services entre eux pour créer des modèles de services complets.

Cas pratique

Illustrons maintenant tous ces points par un cas concret tiré du monde réel. Un distributeur industriel cherche à optimiser sa chaîne logistique, en réduisant le délai entre le passage d'une commande et sa livraison à moins de deux heures. On retrouve dans cette volonté le souhait de rapprocher les processus d'une logique fil de l'eau, événementielle, qui accélère et optimise leur exécution.

Ces processus sont normalisés à l'échelle de l'Europe, pour fournir une base de consolidation de l'exécution du processus au niveau du groupe, mais les spécificités locales sont prises en compte et adressées dans les procédures. On retrouve ici la séparation entre l'interface du processus, son schéma public, et l'implémentation de la procédure, au niveau de filiale, forme de schéma privé.

Le système cible est composé d'un bouquet de services et décrit une SOA. La plate-forme est architecturée sur des Services Web et animée par un serveur d'orchestration. Les services identifiés sont de six catégories très disparates dans leur grain et dans leur description :

- Les **services d'accès au système legacy**, comparables à un connecteur qui accéderait aux données et aux transactions du mainframe. On se situe à une granularité variable entre le service technique et le service métier ;
- Les **services d'accès et de restitution multi-canal** sont des services techniques transversaux de présentation de l'information ;

- Les services de **gestion des référentiels de données** permettent la propagation, depuis un référentiel de données centralisé, de l'information aux applications candidates ;
- Les **services de chorégraphies** BPEL ;
- Les **services métier** et les règles de gestion (via interfaçage avec un moteur de règles), auxquels font appel les services de chorégraphies ;
- La formation de **processus métier** composites par assemblage de services.

Chacune de ces catégories correspond à un des grands domaines couverts par les architectures orientées services (il en existe d'autres). Il est aisé de constater que leur périmètre et leur granularité sont très hétérogènes. Sans classification, il est difficile d'y voir clair dans l'amoncellement de services et de définir une politique de conception, de réalisation et de monitoring adaptée à chaque catégorie. L'administration correcte d'une SOA passe par la définition de cette catégorisation, nomenclature de services, et des liens possibles via la spécification des assemblages réutilisables (les patterns, pour reprendre une formulation architecturale consacrée) et de leurs règles. Sans cela, le risque est important de conduire à un système où les éléments de grain différent se recoupent et se recouvrent, créant une prolifération d'éléments incompatibles pour laquelle aucune volonté de réutilisation, de mutualisation et de rationalisation ne peut prévaloir.

On voit également que la multiplication des services et des catégories de services produit des impacts directs sur le nombre de cartographies et le nombre d'éléments à cartographier. Les architectures multi-niveaux qui en résultent doivent être maîtrisées.

SOA	
Ce qu'on recherche	Ce qu'on doit surveiller
Élément structurant et organisateur	Manque de vision d'ensemble
Accroît réutilisabilité et mutualisation → pilotage et gouvernance du SI	Prolifération des services et des éléments
Urbanisation itérative, incrémentale, par parties	

On peut donc estimer que des outils dédiés sont nécessaires pour organiser cet ensemble et éviter les effets pervers latents du système. On peut également estimer que ces outils de modélisation et d'administration sont différents de ceux qui seront amenés, sur le terrain, à animer la conduite des opérations. On se retrouve ainsi à gérer des systèmes duaux, qui doivent pourtant communiquer : sans cela, la cartographie « théorique » issue de l'atelier de modélisation sera probablement et en permanence décorrélées de la réalité du terrain. Le travail des organisateurs et des opérationnels sera rendu compliqué par cette désynchronisation, et les deux équipes se renverront probablement la balle dès que des difficultés surgiront. Il est donc important de construire des passerelles entre les deux systèmes : c'est ce que l'on nommera l'accostage Urbanisme ↔ SOA.

4.2 L'accostage Urbanisme ↔ SOA

Le contexte

Le constat initial est le suivant : nombre d'entreprises mènent et ont mené des travaux d'urbanisme ayant abouti à différents résultats : schéma directeur, modélisation des macro-fonctions et activités sur la chaîne de valeur, définition de processus et d'objets métier d'entreprise. Ces travaux doivent justifier d'une finalité pratique et concrète, sans quoi la gouvernance du SI n'est pas réelle et les investissements consentis n'offrent pas les retours espérés.

Il est ainsi important que ces travaux soient en phase avec le terrain pour rencontrer l'adhésion des maîtrises d'ouvrage et des maîtrises d'œuvre. Or, les risques de provoquer une décorrélation existent, et sont nombreux : relatif manque de standardisation des moyens, voire de l'utilisation des moyens, documentation produite par les outils limitée, cartographies « plates » (sans référentiel). L'apport d'un référentiel permettrait pourtant de générer cette documentation ou de fournir des fonctionnalités clés comme l'analyse d'impact ou la génération automatique de diagrammes. Par conséquent, les concepteurs manquent de visibilité, comme de moyens d'action et de pilotage. Les éléments produits se retrouvent difficiles à maintenir et rapidement obsolètes face à un SI en constante évolution.

Par conséquent, il n'est pas étonnant de rencontrer de plus en plus d'entreprises à la recherche de modes d'urbanisation pragmatiques et itérative, qui tiennent également compte de la réduction des durées de cycle : face à un environnement en constant changement, il est difficile d'imposer une urbanisation qui vienne toujours des cercles de réflexion. Le terrain doit avoir son mot à dire et moduler la réflexion en fonction de besoins nouveaux. Les résultats d'une démarche d'urbanisation doivent être rapidement mesurables, sans effet tunnel.

La réponse

Pour y parvenir, il est nécessaire d'établir le lien entre la modélisation et l'application des moyens. Le travail d'accostage consiste à dériver un travail d'urbanisme existant pour lui appliquer une spécialisation pragmatique. Il est donc d'abord important de faire un bilan des travaux d'urbanisme produits pour voir ce qui est réutilisable et mesurer la distance d'accostage permettant d'aboutir aux éléments dont nous aurons besoin pour organiser une SOA. Evidemment, si aucun élément d'urbanisation préalable n'existe, il sera difficile de parler d'« accostage » ; mais les cas où rien n'est réutilisable sont relativement rares. D'autre part, lorsque les travaux d'urbanisme existants sont de trop haut niveau, l'étude d'accostage peut parfois n'aboutir à rien de concret : la distance est trop large entre les deux domaines.

Fort heureusement, on connaît les éléments qui sont nécessaires à la construction d'une SOA. L'idée de l'accostage est donc de piocher les éléments de conception réutilisables dans un référentiel existant, pour travailler sur la filiation des travaux et définir les modalités du lien inverse, du terrain vers l'environnement de conception. Dans le cas d'une SOA, les processus, formats pivot, et, plus généralement, tout élément de modélisation, sont repris (et outillés avec un atelier du marché lorsque ce n'est pas déjà le cas) pour être dérivés en direction de la cartographie et de la mise en œuvre de flux d'échange de données, ainsi que de modèles d'orchestration de services (pour la SOA). Citons ainsi un travail d'accostage, qui a consisté à l'identification et l'établissement des éléments suivants :

- Définition d'objets métier d'entreprise,
- Formalisation de nomenclatures et d'un langage commun pour décrire les fonctions de l'entreprise,
- Cartographie des processus d'échange.

L'étape suivante consista à alimenter le référentiel d'un atelier de modélisation du marché avec les objets définis, pour bénéficier des fonctions d'analyse d'impact et de cartographie. Cela nécessita la personnalisation du méta-modèle de cet atelier, pour le faire coller exactement aux besoins de l'entreprise : modification des attributs de description des objets, création de nouveaux objets et de liens entre ceux-ci... La personnalisation d'un méta-modèle est une opération de première importance, et la facilité de personnalisation devrait constituer, dans tous les cas, un critère majeur dans le choix de ce type de produits.

Par la suite, les efforts ont porté sur la génération d'une documentation adéquate et sur les passerelles d'interaction avec les environnements d'exécution du système informatique. L'export de processus BPEL et leur intégration au sein du référentiel de la plate-forme EAI fut ainsi testée et validée.

Lors de la création d'une SOA, l'accostage porte a minima sur l'échange de trois types d'objets incontournables, à l'aide des standards du domaine : la description des données et objets métier avec XML-Schemas, la description des services avec WSDL, et la description des processus avec BPEL. Pour les autres types d'objet, on utilisera des langages XML, tels que XSLT pour décrire les transformations de format à format. Schématiquement, ces passerelles d'échange entre l'atelier de modélisation et le référentiel des environnements d'exécution sont représentées à la figure 21.

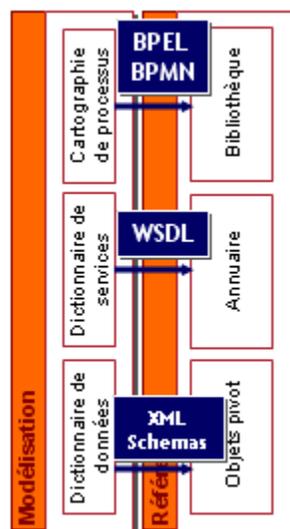


Figure 21 - Passerelles d'accostage basées sur les standards XML

Les processus peuvent être cartographiés à différentes échelles : le formalisme peut être le même, c'est le niveau de détail de la carte qui diffère. Chaque niveau reste nécessaire à une bonne compréhension de l'ensemble, mais tous les cartes ne sont pas destinées à passer du mode « descriptif » au mode « exécutable » en tant que composant technique du système

informatique. Les processus techniques automatisés et les chorégraphies font partie des éléments exécutables. La hiérarchie des services, leur nomenclature et leur catégorisation peuvent également être transférées à l'annuaire de services, si la structuration de celui-ci le permet.

Dans tous les cas, pour répondre aux enjeux qui naissent de la multiplication des services et des niveaux de services, dans le but de maîtriser et de piloter réellement la SOA, la véritable administration s'opèrera au niveau de l'atelier de modélisation et de conception. Les liens bidirectionnels avec le serveur d'exécution sont indispensables pour privilégier flexibilité et réactivité.

4.3 Modes de construction de la SOA

Si une exigence de pragmatisme et d'application naît au niveau de l'urbanisation des systèmes d'information, celle-ci doit encore cependant se vérifier sur le terrain.

Cas pratique

Une grande entreprise du secteur industriel souhaite urbaniser son système d'information en support de l'optimisation de ses processus. Le processus est l'élément pivot à partir duquel toute la stratégie de construction du système cible s'élabore, et il se construit par assemblage de différents niveaux de services métier et techniques.

L'entreprise a conscience que son besoin SOA est large et recouvre des fonctionnalités d'intégration et d'orchestration, d'éléments synchrones et asynchrones. Elle lance ainsi une étude de qualification du système cible idéal, puis une étude de choix pour déterminer le meilleur candidat. Cette étape franchie, on passe à la construction d'un socle minimal destiné à outiller le premier projet pilote. Ce n'est que dans un deuxième temps, après la mise en œuvre du premier projet, que le Centre de Compétences SOA est constitué et les outils choisis.

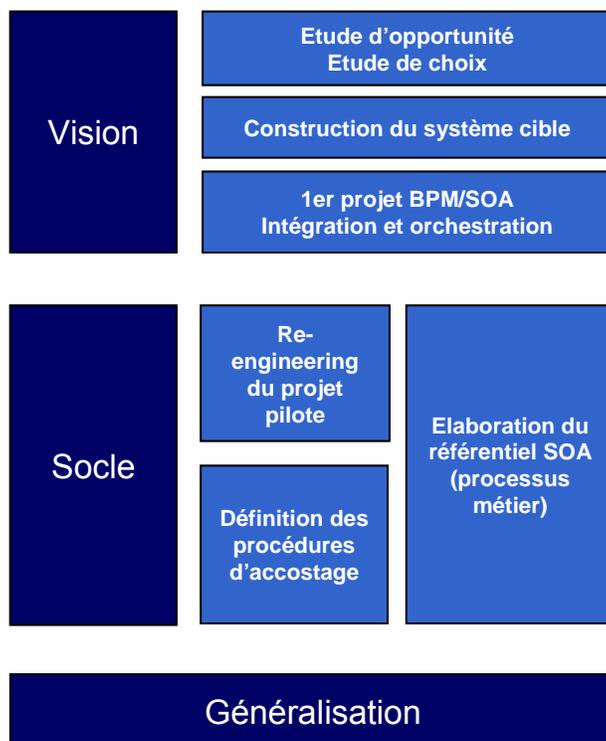


Figure 22 - Méthode de construction itérative du SOA

Ce processus s'explique par le fait que les modes d'organisation et les technologies des SOA sont complètement nouveaux pour bon nombre d'entreprise, et qu'en la matière, il n'y a pas de vérité absolue : il faut tester avant d'approuver. Ne serait-ce que pour les questions d'organisation, la création possible d'un Centre de Compétences ne se traduira pas de la même façon dans deux entreprises différentes. Il n'y a pas réellement de modèle à préconiser, mais une matrice de fonctions et de processus collaboratifs de laquelle l'entreprise peut s'inspirer pour comprendre ce qui convient à ses besoins et à sa structure et le mettre en œuvre.

Le Centre de Compétences SOA

Cela est dû au fait que la construction et l'exploitation efficace d'une SOA dépendent de l'autonomie de ce Centre de Compétences dans le pilotage du système d'information. Tout projet qui débute doit savoir à quelles règles obéir pour se glisser dans l'organisation existante et respecter les modes de fonctionnement définis, de même qu'il doit connaître la nature et le type des services qu'il peut être amené à (ré)utiliser pour ses propres besoins. Face à l'organisation Projet toute puissante, ce mode d'organisation, inhabituel et spécifique, demande à l'entreprise une période d'adaptation. Comme le package technologique sur lequel cette organisation repose est également récent et complexe, tout cet ensemble peut demander un premier projet pilote durant lequel l'entreprise tâtonnera et affindra sa perception. Ce n'est qu'ensuite que le véritable socle sera lancé, via la définition des procédures d'accostage et la construction du véritable référentiel SOA et de sa nomenclature de services, pour être généralisé ensuite, industrialisé, appliqué à l'ensemble des projets candidats.

Construire une SOA n'est pas une démarche que l'on ne peut souhaiter dérouler que sur un projet unique. C'est un choix d'urbanisation qui vise à privilégier la réutilisabilité des services et

l'homogénéité du système d'information, sur un plan fonctionnel comme technique. Ce sont des enjeux d'entreprise, un assemblage structurant, pour lequel il faut tolérer quelques approximations de lancement ; car il est souhaitable qu'un tel modèle soit généralisé, si l'on veut qu'il délivre tout le retour sur investissement promis.

Cette démarche itérative permettra également de se prémunir des approches trop dogmatiques qui pourraient conduire à une utilisation abusive des services et à un *big bang* déraisonné. Un système d'information *full SOA* doit constituer une situation idéale cible vers laquelle on tend, règle d'urbanisation forte devant guider la démarche pour en réussir l'implantation. Il ne s'agit pas de forcer le passage vers un modèle tout service, mais de la faire comprendre et accepter.

C'est donc un nouveau modèle qui s'installe et se déploie et demande son lot de zéloteurs pour le promouvoir et le défendre. Comme toujours, après avoir validé la démarche, il faudra que l'entreprise identifie des personnes dédiées à la communication sur ces sujets, capables d'intervenir sur les phases amont des projets en cours de conception pour généraliser la vision dans la mesure où celle-ci est compatible avec les contraintes techniques du dit projet.

Le travail sur l'existant

Le travail d'exposition de services ne concerne pas que les nouveaux projets et les nouvelles technologies: il s'adresse aussi à l'existant. Certaines applications peuvent ainsi être étudiées sous l'angle des services qu'elles peuvent exposer, et ce, de façon autonome ou bien en conjonction avec les souhaits des urbanistes. Il s'agit d'ouvrir de nouvelles interfaces, de type Services d'accès aux applications, depuis un parc légataire, afin de sortir l'application de son flot existant. Une forme de rénovation de façade, en quelque sorte, pour uniformiser un modèle et rendre le parc fonctionnellement homogène et interopérable, quand bien même il reste technologiquement très hétérogène.

On voit ainsi apparaître un travail d'un nouveau genre, à mi-chemin entre la conception et le reverse engineering, faisant participer urbanistes, architectes, responsables d'application et expert des technologies sur lesquelles celles-ci sont conçues. Lorsque l'application considérée n'expose pas nativement un ensemble d'API, la nature du travail d'exposition des services peut demander un travail conséquent d'adaptation de l'existant et s'avérer assez intrusif.

Un projet (appelons le Projet A) qui aurait besoin d'accéder aux services d'un parc légataire pourrait donc entraîner, par ricochet, l'exposition de certains services non encore ouverts depuis ce parc légataire. Ce travail peut difficilement être imputé au budget du projet A. Il fait partie de la cible d'urbanisation : le budget du Centre de Compétences devrait être sollicité. Pour autant, il faut que celui-ci dispose de l'autonomie suffisante pour démarcher les responsables d'applications du parc légataire et lancer les travaux de rénovation adéquats. L'organisation doit suivre, sinon la construction de la SOA reste limitée et ne tient pas ses promesses.

Pour toutes ses raisons, la question du financement d'une Architecture Orientée Services d'entreprise doit être réglée dès les premiers pas de l'initiative, et apporter des réponses au cas cité ci-dessus, comme à bien d'autres. Par exemple, qui couvre l'acquisition de nouveaux serveurs pour faire face à la montée en charge de l'ensemble du système et aux besoins de haute disponibilité ?

5. Les outils de la SOA

La maturation rapide mais encore inachevée des standards et des technologies n'a pas empêché un ensemble d'acteurs de l'industrie informatique de se positionner, dès l'arrivée même des Web Services, sur un marché qui est de loin devenu le plus visible et le plus porteur à l'heure où nous écrivons ces lignes. Comme sur tout marché en construction, on peut s'attendre à quelques remaniements significatifs au fil des mois ; néanmoins, vous constaterez que les principaux acteurs ont une assise et une ancienneté suffisantes pour qu'aucun doute ne vienne peser sur leur crédibilité comme sur leur pérennité.

5.1 Le modèle technico-fonctionnel

De façon macroscopique, le modèle des orchestrateurs SOA peut se représenter comme l'indique la figure ci-dessous :

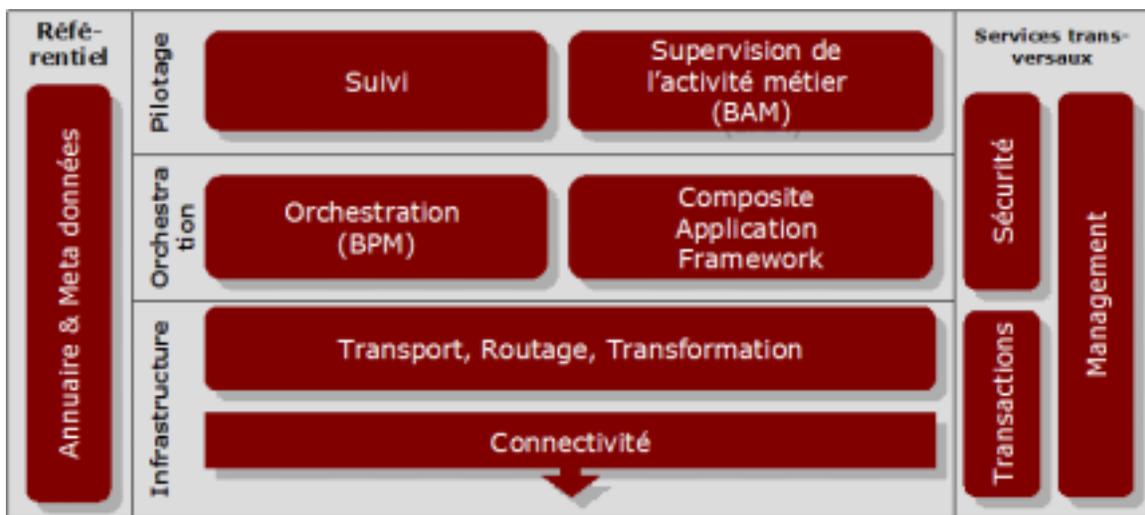


Figure 23 - Modèle technico-fonctionnel

Il est important de noter qu'en fonction des produits, tout ou partie de ce modèle est couvert. Les fournisseurs de plates-formes auront tendance à couvrir plus ou moins bien l'intégralité du modèle, là où les *pure-players* se spécialiseront sur certaines briques. Les descriptions de produits que vous trouverez plus loin dans ce chapitre éclairciront le périmètre respectivement couvert par l'un et l'autre des acteurs identifiés.

D'autre part, le modèle ci-dessus est celui des serveurs d'exécution SOA. Les fonctions de conception et de modélisation de SOA, présentées au paragraphe *L'accostage Urbanisme* ↔ SOA.

Sur la figure ci-dessus, cinq blocs se dégagent :

- Le **référentiel** de services et de méta-données ;
- L'**infrastructure** de transport et de connexion, couche basse du modèle ;
- Les services d'**orchestration**, couche intermédiaire d'organisation de la plate-forme et d'enrichissement métier ;
- Les services de **pilotage**, couche haute de suivi et de supervision ;

- Les **services transversaux**, appliqués à l'ensemble des couches du modèle.

Le référentiel

C'est LE point vital du système, celui qui gère les méta-données de la plate-forme et la catégorisation des services, qui assure le lien entre consommateurs et producteurs de services, qui assure la passerelle entre l'environnement de modélisation et l'environnement d'exécution. Certains pure-players, tels que Systinet, se sont spécialisés dans la fourniture de ce composant incontournable, accolé à des plates-formes d'exécution tierces.

L'infrastructure de transport et d'invocation

Elle couvre l'accès aux services et aux données, ainsi que le transport et le routage des informations à collecter ou à diffuser, généralement au format XML. On y trouve un ensemble de technologies et de protocoles. Le plus fréquemment rencontré jusqu'ici est l'échange de messages SOAP sur HTTP, mais certains éditeurs utilisent des formats propriétaires, comme SAP avec le protocole XI 3.0 (un format SOAP élargi à la prise en compte des attachements⁴). L'aspect asynchrone est couvert par un middleware de type MOM généralement fourni avec l'offre. WSDL 2.0 prévoit des mécanismes asynchrones dans ses spécifications, mais ne fait pas partie de Basic Profile et n'est donc pas encore, à ce titre, largement implémenté.

Il est d'autre part nécessaire de noter que ces produits sont majoritairement des activateurs de SOA construits à base de Services Web. Les systèmes informatique des entreprises n'étant pas tous prêts à s'architecturer autour de cette seule technologie, il est nécessaire d'inclure aux offres des fonctions de connexions aux sources de données via des connecteurs applicatifs plus traditionnels du monde de l'intégration EAI, ce qui tend une fois de plus à rapprocher, de façon tout à fait logique et prévisible compte tenu du périmètre de ces produits, l'intégration EAI de l'orchestration SOA, au sein d'un grand ensemble unifié d'intégration de données, de services et de processus.

Les services transversaux

Ils regroupent les services applicables à l'ensemble des couches de la plate-forme SOA, parmi lesquels les plus notables sont ceux dédiés à la gestion de la **sécurité** et aux **transactions** (où mécanismes de type XA et procédures de compensation cohabitent pour fournir le plus large panel). Les produits se distinguent surtout à l'heure actuelle par leurs différences dans leur niveau d'implémentation des initiatives et standards liés à la gestion de la sécurité et des transactions, présentés dans la première partie de ce document.

On y trouve également les **services de management/administration** de la plate-forme, permettant la gestion du cycle de vie des messages, des services et des processus, de leur conception jusqu'à leur exploitation, en passant par les procédures de déploiement et la gestion de versions qui en résulte. Ces services sont croisés avec certains services de sécurité, dans la gestion des droits d'accès et des habilitations, éventuellement via un interfaçage avec un annuaire LDAP d'entreprise.

L'orchestration de processus métier

⁴ D'où le besoin déjà mentionné de normaliser les modes d'invocation (WS-Basic Profile)

La brique **Orchestration** désigne les moyens d'assembler des services pour décrire des processus complexes, transversaux, comprenant notamment des procédures de compensation. Les processus qui y sont décrits assemblent les processus techniques de la couche inférieure, donnent un sens fonctionnel à la représentation, apportent des fonctions de gestion des interventions humaines avec des algorithmes d'attribution des activités et des fonctions de gestion de l'état suspendu d'un processus. En fonction de l'étendue des fonctions fournies par ce gestionnaire, on le catégorisera davantage comme chorégrapheur de processus techniques ou comme gestionnaire de processus métier (BPM). Ce moteur d'exécution devra bénéficier d'un environnement de réalisation graphique capable de supporter le mode projet en équipe. Il en ressort qu'au cours d'une phase de choix d'un produit, le profil de l'orchestrateur doit être soigneusement défini.

La brique **Composite Application Framework** fournit les moyens d'assembler des applications composites, via notamment des outils de portail et des fonctions de *workflow* humain incluses au gestionnaire d'orchestration. Il s'agit d'outils de développement qui étendent le périmètre couvert par ce gestionnaire pour le rapprocher de celui d'une plate-forme d'infrastructure APS où l'on trouve également des fonctions liées notamment aux serveurs d'application ou aux portails d'entreprise.

Les services de pilotage

Ils couvrent les moyens mis à disposition des utilisateurs pour piloter et superviser la plate-forme, techniquement et fonctionnellement. Nous laisserons de côté la supervision technique, généralement dédiée à des outils tiers (consoles centralisées comme HP Openview ou BMC Patrol), pour nous concentrer sur les outils de suivi et de supervision.

Par **suivi**, on entend l'ensemble des consoles et outils permettant de tracer le cycle de vie des données, l'ensemble des invocations de services et l'exécution de processus, de relever les anomalies, d'entreprendre des reprises sur incidents et des procédures correctives, et d'opérer des requêtes multi-critères sur l'ensemble du champ des opérations.

Par **supervision de l'activité métier**, on trouve l'ensemble des outils permettant la construction d'indicateurs clés de performance, éventuellement temps réel. Ce sont des outils traditionnellement rattachés à un domaine de l'informatique connu sous le nom de BAM (Business Activity Monitoring). La mise en corrélation d'événements techniques et métier avec analyse d'impact, de même que l'analyse prédictive à partir de comportements déterministes déjà constatés, sont des fonctions de plus en plus recherchées. Leur présence sortira l'orchestrateur SOA d'une dimension essentiellement IT et départementale, et confèrera à l'initiative dans son ensemble une meilleure visibilité dans l'entreprise.

5.2 Les fonctionnalités à suivre

Cette partie met l'accent sur certaines des fonctionnalités clés localisées dans chacun des blocs présentés ci-dessus et sur lesquels il est nécessaire de mettre l'accent, chaque produit pouvant présenter des façons très hétérogènes de couvrir ces fonctionnalités.

Le respect des standards

Lorsqu'il s'agit de comparer des produits, un des premiers axes d'évaluation est le niveau d'implémentation des différents standards du monde des Services Web. Ceux-ci sont

relativement nombreux et sont parfois encore en cours de finalisation. On peut donc s'attendre à ne pas les trouver immédiatement implémentés dans les produits du marché, ou à les trouver implémentés avec des niveaux de maturité différents, voire avec des extensions propriétaires. Le premier niveau de qualification d'une offre consiste ainsi à étudier la liste des standards proposés pour chacun des postes (fondations, sécurité, transactions, processus...)

Les transformations

La transformation de format à format, fonction-clé des plates-formes d'intégration EAI, se retrouve également en bonne place dans les plates-formes d'orchestration SOA et contribue au rapprochement des deux catégories de produit. Le cas d'urbanisme, cité plus haut dans ces pages, de migration itérative d'applications, en témoigne. La qualité et la complétude des fonctionnalités de transformation devront être surveillées, même si, à l'heure actuelle, pour la majorité des produits, le niveau atteint témoigne d'une forte maturité.

Les transactions

Dans l'orchestration SOA, la transaction distribuée entre plusieurs environnements ou plusieurs applications est bien souvent perçue comme devant être basée sur des mécanismes type XA. Malgré tout, on s'aperçoit à l'usage que ce type de mécanisme est difficile à mettre en œuvre, parce qu'il mobilise trop de systèmes simultanément, qu'il n'est pas toujours très efficace, et qu'un *rollback* global peut parfois s'avérer catastrophique. On aura donc tendance à rechercher les produits qui proposent des procédures de compensation, dont les mécanismes sont par ailleurs en voie de standardisation au travers de la spécification du langage WS-Business Activity.

La sécurité

La sécurité peut encore bien être le parent pauvre des efforts de standardisation, il n'en reste pas moins que c'est le domaine des Services Web sur lequel les annonces se multiplient actuellement (ainsi du partenariat entre Sterling Commerce et Entrust) et qui génère probablement le plus de mouvements. Au-delà des standards construits ou en cours de construction, il conviendra d'être attentif à l'ensemble des fonctionnalités nativement pourvues par les plates-formes, sans perdre de vue que certains acteurs, comme Oblix, se sont spécialisés sur ce thème et sont interfaçables avec l'ensemble des produits présentés ici.

Le workflow humain

L'intervention des utilisateurs dans le cours d'un processus, technique ou métier, est devenu une fonctionnalité indispensable dans les attentes des entreprises. Celle-ci peut être couverte avec plus ou moins d'intégration au reste de la plate-forme et avec des fonctionnalités plus ou moins complètes. La gestion d'une corbeille de tâches (*to-do list*), de différents niveaux de responsabilité sur les processus (avec escalade vers des niveaux de responsabilité supérieure), l'affectation de tâches à des utilisateurs ou des groupes d'utilisateurs en fonction de leur plan de charge, la reprise sur incidents... sont des fonctions qui ne sont pas toutes couvertes avec le même niveau de complétude et demandent un examen sérieux.

Environnement unifié

Au niveau du service unitaire comme du processus métier, de l'administration du référentiel comme de l'orchestrateur et de la supervision, il est largement préférable de disposer de produits qui proposent une vision d'ensemble du cycle de vie des projets, plutôt que d'avoir à jongler entre plusieurs outils. Aujourd'hui, c'est peut-être là que le bât blesse le plus et qu'il convient d'être attentif, et ce, pour les produits au périmètre le plus vaste, proche de la plate-forme d'infrastructure complète, comme pour les *pure-players*. En effet, dans le premier cas, les plates-formes ensemblistes sont encore en cours de maturation alors que, dans le deuxième cas, le périmètre limité des produits oblige à des partenariats qui, comme le veut l'expression, ne sont pas toujours « sans couture ».

Pour prendre un exemple au niveau de l'orchestrateur, la réalisation des activités de workflow humain gagnera à être totalement couverte par l'environnement de développement, plutôt que déléguée à des modules tiers de la plate-forme.

Gestion des versions

Un orchestrateur SOA étant amené à gérer simultanément des services et des processus à base d'assemblage de ces services, il doit offrir une capacité à gérer intelligemment l'exposition de multiples versions de ces deux types d'objet, ceci afin de gérer la complexité qui résulte de leur multiplication et que nous avons déjà évoqué dans la partie consacrée à l'accostage Urbanisme ↔ SOA.

En effet, la modification d'une version de service peut rester invisible pour le processus qui l'utilise, pour peu que l'interface du service, dans sa nouvelle version, demeure inchangée pour le processus. La figure 24 représente cet état de fait.

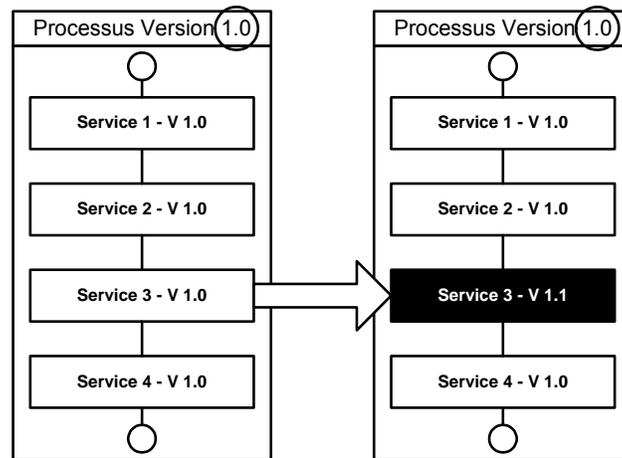


Figure 24 - Changement de version de service

Dans ce cas, la version de processus ne change pas car l'interface qu'il expose au monde extérieur n'est pas modifiée. On considèrera que le processus est impacté si sa définition change. La figure 25 représente un changement de version dû à une modification de la définition du processus (insertion d'une condition).

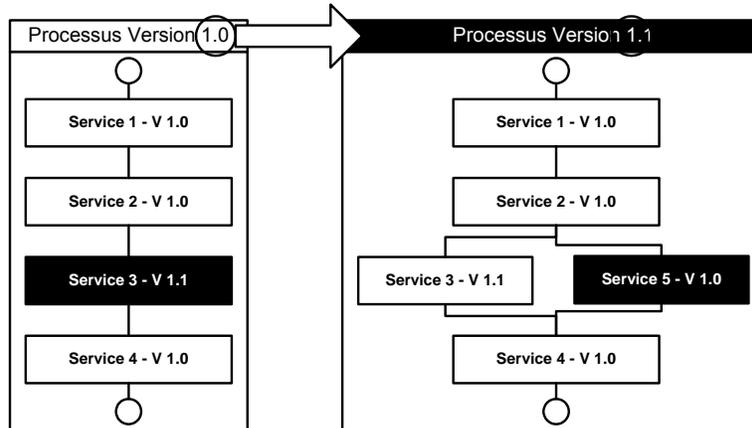


Figure 25 - Changement de version de processus

Les fonctionnalités recherchées doivent couvrir le processus de conception, de réalisation, de mise en production, d'approbation de chacune de ces étapes et de publication dans l'annuaire, au sein d'un workflow adapté. Certains produits de gestion de référentiel, tel que XML-Canon de Tibco, proposent de tels mécanismes.

La gestion de contexte

Au-delà de la gestion de version, il faut bien constater que la même version de service sera utilisée différemment en fonction de l'acteur qui l'invoque : application interne de l'entreprise, partenaire externe, client... si tous les services ne pourront être utilisés par tous les acteurs, ceux qui auront l'habilitation de le faire ne le feront pas dans les mêmes conditions, et passeront au service des informations identifiant l'émetteur. La gestion de contexte introduit une flexibilité nécessaire, que la gestion de version seule ne permet pas, ou seulement de façon réductrice et rigide. On trouve ainsi des logiciels spécialisés dans la gestion de contexte (à ce titre, citons le logiciel EBX.Platform d'Orchestra Network) mais dans tous les cas, la gestion de contexte doit être intégrée dès la conception des services et demeure un élément de méthode important à considérer.

Analyse d'impact

Les fonctionnalités d'analyse d'impact qui existent au niveau de l'atelier de modélisation et de cartographie des services doivent également se retrouver au niveau de l'environnement d'orchestration, pour les opérations chirurgicales tactiques qui demandent une forte réactivité court terme, pour laquelle il ne sera pas possible de se retourner vers l'environnement de modélisation.

La gestion des environnements

Le passage des environnements de développements à la certification/pré-production, puis à la production, doit se faire dans le respect des règles de l'art, avec une segmentation claire des environnements et la capacité à exporter/importer des éléments de granularité précise. Un outil qui ne permettrait d'importer/exporter que des blocs du référentiel de services, voire tout le référentiel sans distinction, s'avèrerait inadapté devant le besoin de gérer des versions potentiellement multiples de services et de processus. La structure du référentiel/annuaire sur

lequel repose l'orchestrateur, et les outils qui permettent de l'administrer, définissent ainsi clairement les possibilités en la matière, et doivent être regardées à la loupe.

5.3 Les catégories de produit

Les acteurs du marché des SOA peuvent se répartir en fonction du type d'offre qu'ils proposent. On en dénombre cinq :

Catégorie	Description et positionnement
Les <i>pure-players</i> SOA	On trouve ici des sociétés de taille modeste, poissons pilotes qui ont défriché le secteur lorsque celui-ci se développa fortement fin 2001/début 2002, comme par exemple Cape Clear et WestBridge aux Etats-Unis. On peut ajouter à cette liste Amberpoint, The Mind Electric, racheté par webMethods, Collaxa, racheté par Oracle, Systinet et Oblix, offres qui nous détaillons dans les pages suivantes.
Les ESB (Enterprise Service Bus)	Produits dédiés à l'orchestration de services sur la base des standards du domaine. Les principaux acteurs sont Fiorano et Sonic Software.
Les plate-formes d'intégration EAI (Enterprise Application Integration)	Ils élargissent leur socle d'infrastructure à de nouvelles fonctionnalités, pour se diversifier en adressant un périmètre d'intégration élargi aux services. On retrouve ici principalement SeeBeyond, Tibco et webMethods. L'extension de leur offre ne s'arrête à la couverture des SOA : elle lorgne ouvertement vers la construction d'une APS (Application Platform Suite), choix majeur d'évolution.
Les orchestrateurs BPM	Les orchestrateurs BPM fournissent les couches hautes et transversales du modèle SOA (annuaire, orchestration, management, suivi et supervision BAM) et délaissent les couches basses. Le modèle qu'il propose est incomplet et n'est pas suffisant pour construire une SOA, mais ces produits peuvent être considérés si l'entreprise dispose déjà par ailleurs des couches basses, notamment du middleware asynchrone, car la compatibilité est bonne. Nous présentons ici Intalio, dont la conception des produits est particulièrement adéquate à la construction d'une SOA.
Les fournisseurs d'APS (Application Platform Suite)	Grands éditeurs généralistes du domaine (BEA, IBM, Microsoft, Oracle, SAP), qui assemblent des éléments logiciels fournis depuis plusieurs années mais proposés selon une vue d'ensemble architecturée autour des services, pour construire des plates-formes complètes capables d'adresser l'intégralité des besoins d'exécution, d'exploitation et d'administration fonctionnels et techniques.

Les pure-players SOA

Dès 2000 et l'émergence du modèle des Services Web autour du protocole SOAP sont venus se positionner des acteurs spécialisés. Amberpoint, CapeClear, Collaxa, Oblix, Systinet, The Mind Electric, WestBridge... sont tous venus proposer une solution ou un ensemble de solutions apportant une ou plusieurs pièces d'un vaste puzzle. Ces positionnements précoces permettent d'identifier les directions vers lesquelles il est incontournable d'avancer : encapsulation de composants Java et .NET sous forme de Web Services, orchestration BPEL, management de l'exécution des processus, sécurité d'accès aux services, annuaire UDDI...

L'avenir de ces sociétés est bien souvent d'être rachetées par des acteurs au positionnement déjà affirmé sur d'autres secteurs dans leur quête de se positionner sur le marché considéré. Le marché des SOA ayant affirmé sa pérennité, on a vu ainsi notamment webMethods racheter The Mind Electric et Oracle, le grand Oracle, étonnamment et trop longtemps absent, racheter

Collaxa et son orchestrateur BPEL. Vous trouverez la présentation de certains de ces produits dans les pages suivantes.

Les ESB

C'est le Gartner Group qui a identifié ces produits comme un segment de marché novateur dédiée à une réponse bien précise : l'orchestration de services, et notamment de Services Web, par le renforcement d'une couche de messagerie asynchrone. Ainsi, voici, aux cotés des pure-players du domaine, les premiers activateurs de l'approche SOA sur le marché : un bus de transport asynchrone de messages, sur lequel viennent s'enficher des services communicants. Ce faisant, le bus devient un moyen de mettre en relation consommateurs et producteurs de services par couplage lâche, sans perte de données.

Ne serait-ce leur relative jeunesse, ces produits ont tout pour plaire et s'imposer :

- d'abord, précisément, leur insolente jeunesse, qui, dans notre industrie avide de nouveautés, les pare d'atours à faire pâlir d'envie les « ancêtres » de l'EAI ;
- ensuite, leur volonté farouche de ne reposer que sur les standards établis du marché. On trouvera donc dans ces outils toute la panoplie actuelle des standards de l'Internet et des Web Services ;
- enfin, un montant d'investissement plus faible, tout au moins pour les coûts de licence, qui leur confère également parfois un positionnement d'« *EAI light* ».

Quelques freins de plusieurs ordres se sont cependant manifestés pour empêcher ces produits de réellement prendre leur envol :

- La jeunesse attire tous les regards, et les ESB furent très rapidement placés sous les feux de la rampe ; mais elle témoigne également d'un manque de maturité qui peut se révéler incompatible avec un objectif aussi ambitieux que de bâtir une SOA urbanisée propre à l'alignement métier. Ces produits se sont donc d'abord avérés intéressants pour adresser des besoins d'orchestration départementaux, ou dans des phases de prototypage et de *proof-of-concept*.
- De plus, tout au moins dans les premières versions, certains des standards les plus en vue n'étaient pas stabilisés, et il subsistait même encore quelques points d'ombre sur la définition même des SOA. Cela se traduit par des produits qui se cherchaient encore. Leur positionnement d'EAI light, mid-market, dévolus à des besoins tactiques, est dû au fait que, de par leur arrivée récente, ils ne couvraient finalement qu'une fraction du périmètre des grands EAI, sans pour autant être spécialisés de façon unique sur un autre segment. Comblant leur retard au fil des versions, ils ressemblent désormais davantage à la vision du Gartner Group.

Cependant, dans la mesure où il est inutile de rester assis à pleurer sa jeunesse, les compétiteurs déjà en place trouvèrent un second souffle en réorganisant techniquement et commercialement leurs produits, dans le sens d'une adoption massive des standards, puis en publiant à leur catalogue un sous-ensemble de leur offre sous la bannière d'« Express » (IBM WBI Express, BEA Express, webMethods Express...), à des tarifs d'entrée plus abordables. Ce faisant, ils redynamisent le marché des EAI mid-market, via un package d'un sous-ensemble des fonctionnalités ; l'entreprise peut ainsi évoluer ensuite en fonction de ses besoins vers une

version plus étoffée en terme de fonctionnalités et de technologies, sans remettre en cause le socle existant.

Témoignage du repositionnement de ces produits mais également source d'une certaine confusion, on voit depuis quelques mois certains éditeurs repositionner les ESB comme la couche basse de leur modèle SOA, excluant de facto l'orchestrateur, placé dans les couches hautes. Pourtant, qu'est-ce qu'un ESB sans orchestrateur ? Pas grand-chose, n'est-ce pas ? Et par ailleurs, toutes les sociétés positionnées par le Gartner Group comme acteurs de l'ESB (Sonic, Fiorano...), fournissent un orchestrateur. Réduire l'ESB au seul bus d'échange semble constituer une notion bien trop limitative pour être réellement significative.

Les fournisseurs d'EAI

Jusqu'à présent, nous avons coutume de positionner deux, voire trois approches, dans notre présentation du périmètre EAI :

- L'approche Données correspond à une volonté de disposer d'une information synchronisée ou consolidée pour les sources de données du système d'information.
- L'approche Processus consiste à vouloir conduire, au travers de la plate-forme d'intégration, des processus transversaux qui décrivent le cycle de vie d'un message métier (exemple : une commande, un colis, un titre financier...) au travers du système d'information.
- L'approche Pilotage cherche à positionner des capteurs sur les messages métier transportés, pour y collecter des valeurs ou des jalons permettant d'établir la performance de l'activité de l'entreprise.

La figure ci-dessous représente ces trois approches sous forme de cercles concentriques indiquant le caractère itératif et incrémental des trois approches dans la mise en œuvre d'une dorsale d'échange.

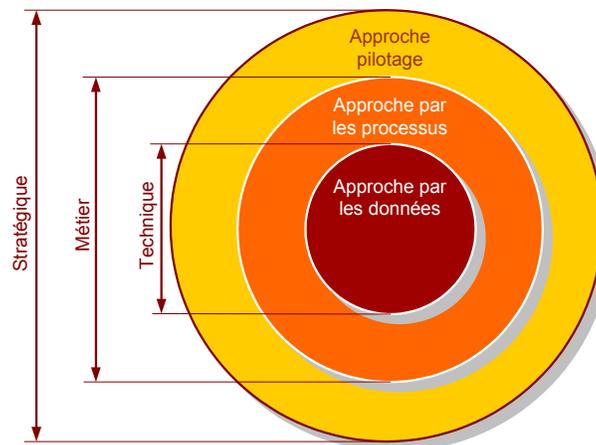


Figure 26 - Les trois approches "traditionnelles" de l'EAI

La réalisation et le déploiement de services ont pour vocation de s'intercaler entre l'approche Données et l'approche Processus :

- Concernant l'approche Données, nous avons évoqué plus haut dans ces pages la volonté de créer, via la mise en œuvre de services dédiés, une couche de services d'accès pour éviter un accès natif au modèle de données du référentiel, masquant également ainsi la structure interne de ce référentiel.
- Concernant l'approche Processus, nous avons vu que celui-ci peut être constitué à partir d'un assemblage de services, définissant potentiellement eux-mêmes un (sous-) processus, ce qui apporte davantage de granularité comme de possibilités de réutilisation et de mutualisation.

Techniquement, cela se concrétise sous deux formes :

- La fourniture d'un connecteur Web Services permettant l'accès à un annuaire UDDI et l'invocation de requêtes SOAP ;
- La capacité à exposer un processus sous forme de service.

Pour bénéficier d'une véritable approche SOA, il faut de plus faciliter l'accostage via les outils d'administration du référentiel ou des passerelles vers des outils tiers. Dès lors, les EAI sont susceptibles d'évoluer vers une approche unifiée EAI/SOA grâce à l'apport des fonctionnalités déjà décrites ci-dessus.

Cela ne remet donc pas en cause la structure même des plates-formes d'intégration EAI. En fait, l'intégration de services est un enrichissement de ces produits, qui témoigne du caractère complémentaire de l'intégration EAI et de l'orchestration SOA, les deux modèles étant bâtis sur le même middleware asynchrone assurant le couplage lâche et la garantie de livraison des informations. L'apport d'un middleware asynchrone est même indispensable. Cette complémentarité s'est notamment traduite par l'acquisition en octobre 2003 par l'éditeur américain d'EAI/B2B webMethods de la société The Mind Electric et de ses produits SOA Glue et Fabric. Cette adoption des SOA via acquisition ou partenariats (pour l'éditeur Tibco), ou encore via un développement maison, n'a pas été interprétée par les analystes comme un reniement ou un renoncement, mais comme une évolution vers une couverture plus large d'une infrastructure dont le périmètre global s'est étendu. La conséquence en est la quasi-disparition du terme EAI dans le discours de la majorité des éditeurs sans que cela ne remette en question les principes mêmes des produits évoqués.

L'orchestration de Services devient ainsi un chaînon manquant entre l'approche EAI orientée Données et l'approche orientée Processus. Elle fait désormais partie du catalogue des principaux éditeurs d'EAI, et l'avenir de ceux qui refusent cette évolution apparaît aussi sombre que celui de ceux qui n'ont pas su évoluer vers le processus à l'orée de l'an 2000.

Les APS (Application Platform Suites)

Les fournisseurs d'APS sont des sociétés implantées depuis plusieurs années, voire dizaines d'années, sur le marché de l'informatique, et proposant une offre souvent très large, allant du métier, avec des progiciels intégrés et/ou verticaux, à la technique, avec différents produits d'infrastructure. Faisant partie du quotidien de notre profession, ces éditeurs bénéficient d'une large base installée : il s'agit de BEA, IBM, Microsoft, Oracle et SAP.

A divers titres, ces éditeurs finissent par se retrouver une nouvelle fois concurrents sur le marché des APS, plates-formes d'infrastructure complètes et de logiciels verticaux qui regroupent

généralement au moins un serveur d'applications, un gestionnaire de portail d'entreprise et un serveur d'intégration/orchestration. Sur ces trois briques de base viennent parfois se greffer un gestionnaire multi-canal/mobilité, un gestionnaire des données de références, un atelier de modélisation de données et de processus, voire une offre métier fondée sur un progiciel de gestion intégré (ERP). On dispose alors d'offres duales complémentaires, l'une métier, l'autre destinée aux systèmes intersticiels d'infrastructure.

L'idée est d'utiliser la seconde en tant qu'activateur et organisateur de SOA, et de restructurer la première en découpant ses modules applicatifs sous forme de services. Les processus hébergés par l'ERP pourront alors également être exposés sous forme de services. Le potentiel de l'offre métier sera alors accru par la création de nouveaux processus personnalisés réalisés depuis la seconde et permettant l'interaction avec des applications/services tiers comme avec des partenaires externes.

Plus le périmètre couvert par l'offre est vaste, plus il sera nécessaire pour l'éditeur qui la commercialise de consacrer du temps et de l'argent à construire les modules, à les consolider avec les modules existants et à y inclure les implémentations des nouveaux langages et protocoles qui ne cessent encore d'apparaître.

On constate ainsi un fossé assez massif entre ce que proposent des pure-players au périmètre produit limité, mais très spécialisés, très complets dans les implémentations proposées, et performants, et des fournisseurs d'APS au périmètre produit complet, mais moins précis dans leurs implémentations, contraints d'adapter en permanence leur socle technologique en mutation aux nouveaux langages (facteur d'instabilité proportionnel à l'envergure de ce socle), et, de plus, souvent tentés d'y inclure des extensions propriétaires. Voilà pourquoi les fournisseurs d'APS, tels qu'IBM et SAP, annoncent volontiers une finalisation de l'ensemble de leur socle pour 2007.

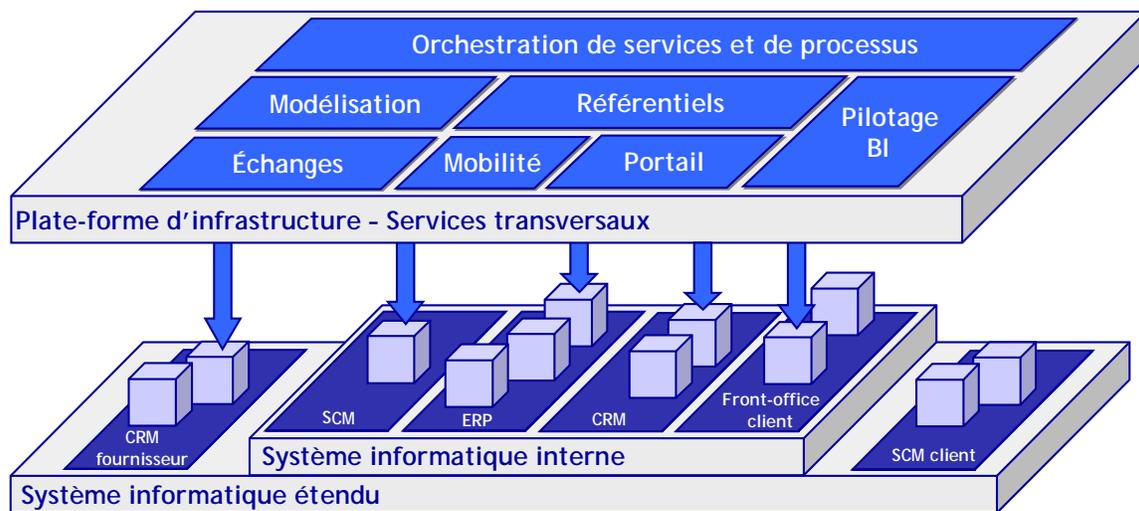


Figure 27 – Plate-forme d'infrastructure transversale

5.4 Les acteurs et leurs produits

Voici, par ordre alphabétique, la liste des acteurs dont vous trouverez le positionnement SOA et les produits correspondant dans les pages suivantes. Nous avons délibérément limité ce tableau à une liste d'acteurs distribués en France ou dans les pays francophones limitrophes.

Editeur	Catégorie
AmberPoint	Pure-player
BEA	APS
IBM	APS
Intalio	Orchestrateur BPM
Microsoft	APS
Oblix	Pure-player
Oracle	APS
SAP	APS
SeeBeyond	EAI → APS
Sonic Software	ESB
Sterling Commerce	EAI
Systinet	Pure-player
Tibco	EAI → APS
webMethods	EAI → APS

AmberPoint

AmberPoint Southern Europe
Imm. Le Meliès
261, rue de Paris
93100 MONTREUIL
Phone: +33 6 24 63 11 13
france@amberpoint.com

Fondée en 2001 d'équipes rompues aux nouvelles technologies et aux architectures distribuées, la société AmberPoint a construit son offre sur le principe de l'organisation des SOA, en réponse à un ensemble de besoins et de challenges clairement identifiés pour maîtriser la complexité inhérente à ces architectures : les transactions, la sécurité, la répartition architecturale des composants sur plusieurs serveurs, la gestion des versions, la gestion des exceptions, et enfin le maintien et le respect de la qualité de service souhaitée.

La solution repose sur les caractéristiques suivantes :

- non intrusive pour assurer l'animation et l'organisation d'une SOA sans impact sur les composants existants ;
- distribuée pour éviter de constituer elle-même un goulet d'étranglement ;
- compatible et implémentée .NET et Java.

Ensemble de produits dédiés à l'exécution et à l'exploitation d'une SOA, AmberPoint s'est associé, pour couvrir un périmètre plus large et remonter vers l'accostage et la modélisation, à Systinet (cf. fiche produit), notamment fournisseur d'annuaire UDDI. Les deux éditeurs proposent un SOA Starter Pack qui accroît la mutualisation et le ROI d'une SOA construite sur leurs solutions respectives.

Les clients d'Amberpoint sont, entre autres, British Telecom, Fujitsu, Mototola, Reuters ou encore Samsung.

Les produits

AmberPoint Express est la version gratuite et limitée de AmberPoint pour les développeurs. Cet outil de suivi permet de mesurer, déboguer et optimiser les fonctionnalités et la performance des Services Web. La génération de fichiers SOAP de test à partir du WSDL des services permet la création automatisée de jeux de test. La mesure des performances s'opère en temps réel via des tableaux indiquant les succès comme les erreurs, permettant de détecter les goulets d'étranglement. Le produit est nativement intégré à l'environnement Visual Studio .NET.

Le produit **AmberPoint Management Foundation** fournit des fonctionnalités de gestion, de supervision et de sécurité. On y trouve également la gestion des versions, le suivi des alertes et la journalisation d'exécution. Le partenariat avec la société Roguewave Software permet d'incorporer dans le framework du code source ou des composants C++ revampés sous forme de Services Web grâce à son framework LEIF (*Lightweight Enterprise Integration Framework*). **Service Level Manager** est dédié à l'affectation dynamique de ressources pour l'exécution des Services Web en fonction des contraintes métier et techniques, afin d'optimiser l'exécution des services et de prévenir les risques ; ce produit opère également des analyses prédictives afin d'identifier les tendances dans la consommation des ressources.

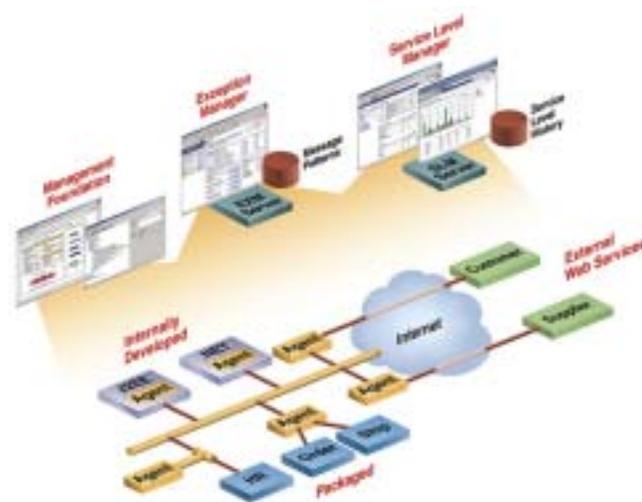


Figure 28 - Modèle architectural AmberPoint

Les briques du modèle SOA

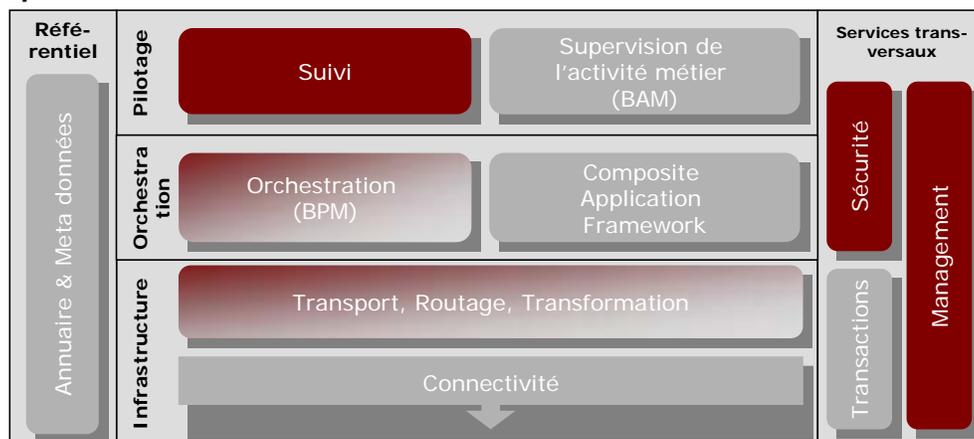


Figure 29 - Le modèle SOA d'Amberpoint

Cette plate-forme d'exécution et de suivi se caractérise principalement par une couverture des couches hautes du modèle, délaissant les couches basses. L'absence de middleware asynchrone semble réserver cette offre aux SOA essentiellement synchrones, mais là encore, le partenariat avec Systinet, via Systinet Gateway, peut ouvrir un accès aux middlewares asynchrones sans remettre en cause l'homogénéité du modèle.

La sécurité constitue un point important : compatible avec WS-Security, la plate-forme implémente XML-Signature et XML-Encryption, et s'intègre aux infrastructures de sécurité existantes (Netegrity).

Le management est dynamique et repose en partie, grâce au produit Exception Manager, sur un système de règles et de corrélations où des actions correctives en temps réel ou des notifications peuvent être automatiquement lancées en réponse à un évènement particulier. Concernant l'exploitation et le suivi technique, la plate-forme sait s'interfacer avec des outils d'entreprise tiers comme Microsoft Operations Manager ou IBM Tivoli. Le suivi est métier, attaché au contenu des documents échangés et non uniquement au contenant.

BEA

BEA Systems SA

Tour Manhattan - 6, place de l'Iris

92095 Paris la Défense Cedex 2

Tél : 01.41.45.70.00,

La société BEA Systems fut créée en 1995 par 3 ex-dirigeants de Sun, autour du produit Tuxedo, moniteur transactionnel qui fit le succès de BEA et de sa vision middleware. Passé ces premiers pas, l'évolution des technologies poussa BEA à racheter en 1998 la société WebLogic et son produit Tenga pour se positionner sur le marché des serveurs d'application : nouveau succès. Suite logique, l'émergence du e-business conduit BEA à doter en 2001 son socle technique de nouveaux composants, d'abord de portail, puis d'EAI, pour délivrer ces fonctionnalités d'infrastructure complémentaires. En 2002, le rachat de la société CrossGain apporta à ces moteurs d'exécution un environnement de développement unifié, notamment dédié à la réalisation de Web Services. En 2003, l'offre WebLogic Platform consolide complètement l'offre de BEA, approche horizontale et unifiée de l'infrastructure d'entreprise, et fait de BEA l'un des fournisseurs majeurs de l'APS. Platform 9 est dans les *starting blocks*.

Le discours de l'éditeur repose essentiellement sur le concept de Liquid Computing, dont l'objectif est précisément l'alignement métier du système d'information pour l'entreprise temps réel, sur la base d'une SOA qui améliore l'adaptabilité et la productivité de l'entreprise. Pour BEA, la réflexion sur les SOA est indissociable d'un travail d'architecture et d'urbanisme. L'important est avant tout de construire une infrastructure complète, basée sur l'utilisation massive de standards, et fortement tournée vers le métier. Services fonctionnels et techniques se côtoient et s'imbriquent pour former un système d'information orienté processus. Dans cette vision, les projets de *revamping* du parc légataire comptent autant que les nouveaux projets : il est essentiel de fédérer les anciennes et les nouvelles applications. Cependant, l'éditeur n'oublie pas d'affirmer que cette évolution progressive du système informatique vers la SOA ne peut s'opérer en un seul gigantesque projet, mais bien progressivement, de façon itérative, en urbanisant par parties.

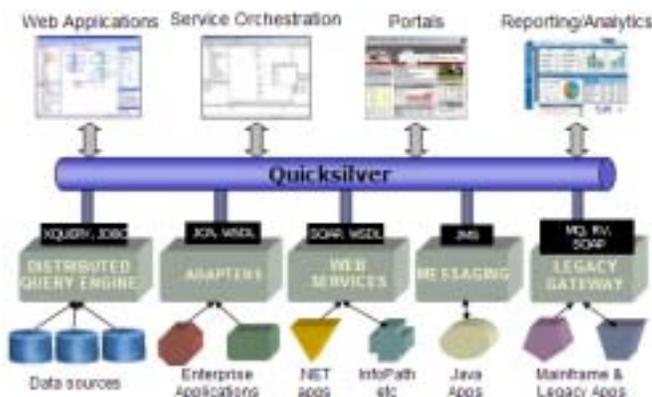


Figure 30 – Le projet Quicksilver

BEA a ainsi lancé le projet QuickSilver, bus d'interopérabilité XML sur lequel viennent s'enficher l'ensemble des services disponibles dans l'entreprise, indépendamment des protocoles de communication dont s'occupe nativement Quicksilver. Cette idée d'indépendance entre l'infrastructure et le fonctionnel est actuellement une tendance forte, que vous retrouverez chez d'autres éditeurs dans les pages suivantes.

Les produits

Platform est construit autour de plusieurs produits présents déjà depuis quelques années dans l'offre de l'éditeur :

- **WebLogic Server** : cœur de l'architecture, le serveur d'application comprend toutes les fonctionnalités techniques transversales nécessaires. Tout besoin spécifique envers des composants de type *portlet* ou BPM doit ensuite guider l'entreprise vers l'utilisation de **WebLogic Portal** (EIP) ou de **WebLogic Integration** (EAI), produits modulaires à acquérir séparément ;
- Signalons **Liquid Data for WebLogic**, base de données virtuelle XML dont la vocation est de créer et conserver des vues de données d'entreprise sous forme de documents XML. Liquid Data est positionné par les analystes comme outil d'EII (création d'un référentiel de données d'entreprise virtuel). Capable de générer et stocker les vues demandées, quel que soit l'emplacement physique des données, il sera demain également capable de travailler en mise à jour. Chaque application peut ainsi travailler en accédant aux données qui lui sont propres, données non centralisées mais regroupées sous forme de vues, consommées par les applications candidates. Le modèle évènementiel est ainsi élargi à la production de l'information et non uniquement à sa consommation ;
- **WebLogic JRockit** est une machine virtuelle Java optimisée pour les plates-formes Intel ;
- Enfin, **WebLogic Workshop** et WebLogic Workshop Framework sont simultanément environnement de développement unifié pour toute l'offre et environnement de packaging et de déploiement des applications.

La figure ci-dessous reprend ce découpage entre les produits :

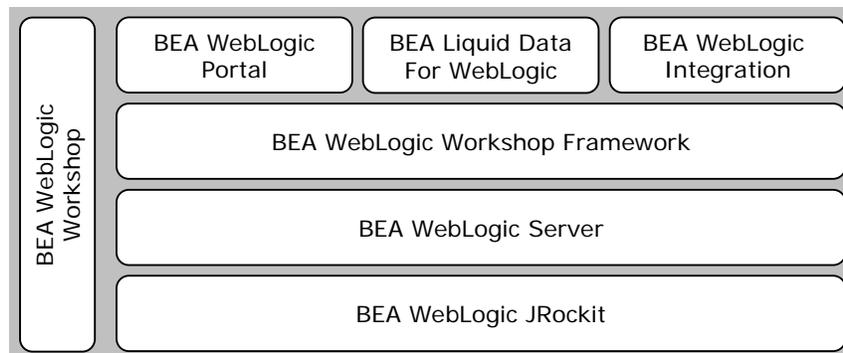


Figure 31 – Les produits de l'offre BEA

Mi-2004, BEA a regroupé un certain nombre de fonctionnalités de ces différents produits sous une nouvelle déclinaison de son serveur d'application nommée BEA Weblogic Server Process Edition. Cette mouture contient les fonctions permettant la construction et l'orchestration de services SOA. D'un point de vue technique le produit reprend le serveur d'application J2EE, enrichi d'un outil de BPM, d'un outil de transformation de données, de connecteurs et d'une console d'administration. Server Process est un outil intermédiaire moins complet que la suite complète Platform et destiné à un usage beaucoup plus réduit.

Les briques du modèle SOA de BEA

BEA proposant une plate-forme, l'ensemble du modèle SOA est couvert, avec une petite réserve cependant concernant la partie BAM, différente de celle de ses concurrents :

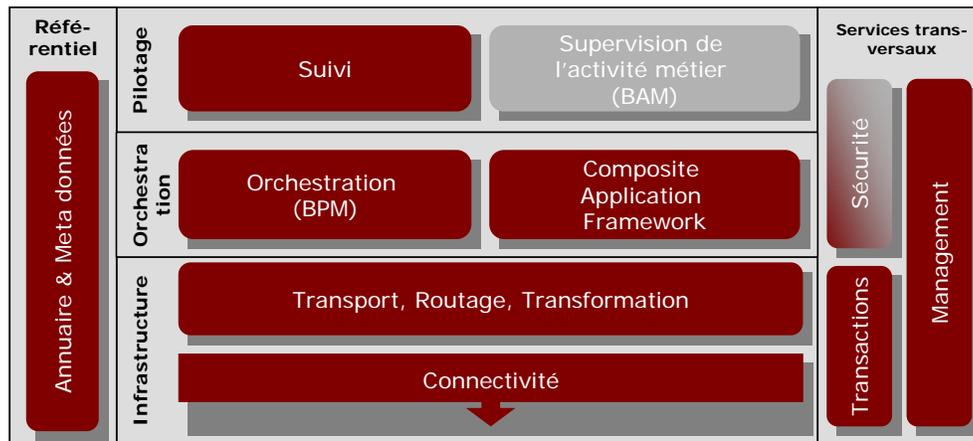


Figure 32 – Le modèle SOA de BEA

En matière de BAM, BEA a ainsi annoncé récemment un partenariat avec BMC Software pour faciliter l'intégration entre BMC Service Impact Manager et BEA Business Process Management et destiné au management d'une SOA (BSM pour Business Service Management) et à l'analyse prédictive.

Par ailleurs, l'éditeur appuie sur les capacités de développement que son offre est capable d'offrir. Celles-ci se matérialisent d'abord dans l'APS : en fournissant une plate-forme complète, on dispose d'emblée d'une capacité à accélérer les développements, et ce dès le premier projet, pour des profils de développeurs bien moins experts qu'auparavant.

Les services d'annuaire sont compatibles LDAP et UDDI V1 et V2. La définition des processus est compatible BPML. De façon générale, l'offre n'est pas consommatrice des derniers standards en vogue et attend leur stabilisation avant de les implémenter.

La gestion du cycle de vie des projets, avec notamment les procédures d'approbation avant validation et déploiement, est couverte.

Les briques du modèle SOA de Fiorano

Le modèle SOA de Fiorano est typique de celui d'un ESB, qui se concentre sur les couches basses et sur l'orchestration.

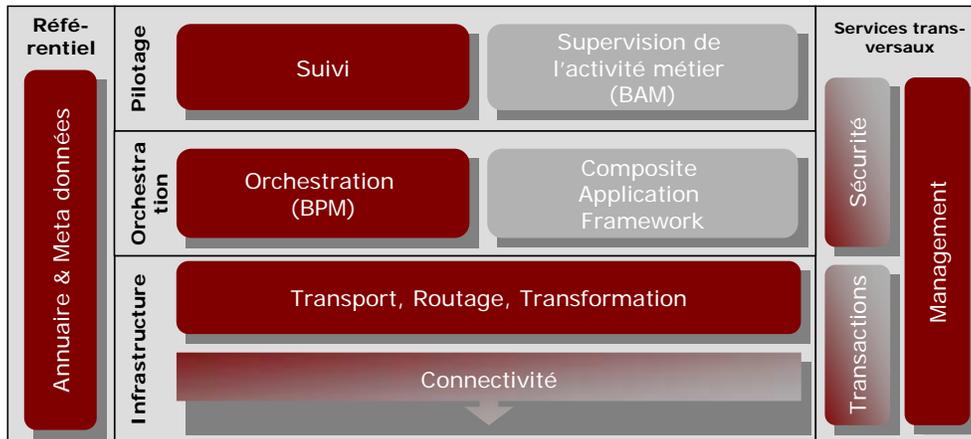


Figure 34 - Le modèle SOA de Fiorano

Fiorano assure une couverture complète de la couche basse grâce au middleware Fiorano MQ et un ensemble de connecteurs applicatifs : connecteurs fichiers, bases de données, progiciels.

Un produit, Fiorano Services et Security Manager, est dédié à la sécurité des échanges et de la plate-forme, en partageant les définitions et droits existants dans des annuaires d'entreprises tiers.

L'orchestrateur constitue le cœur de l'offre. L'ensemble des messages échangés et des services invoqués fait l'objet d'un suivi complet. A fin de résolution rapide d'incidents, les interventions humaines sur le cours des échanges sont prévues.

IBM

IBM France
2 avenue Gambetta
Tour Descartes - La Défense 5
92066 Courbevoie – France

Tél : 0 810 835 426

IBM est leader mondial des services et des technologies de l'information et fait de l'*e-business onDemand* le moteur de son développement, via la réalisation et la commercialisation de solutions globales : matériels, logiciels, services, conseil et financement. Une entreprise *onDemand* est une entreprise pour laquelle les processus métier sont intégrés de bout en bout, en interne comme avec tout partenaire. Les principes de la SOA font ainsi clairement partie du positionnement *onDemand*.

IBM place le débat SOA au niveau métier et IT : en optimisant l'infrastructure, on assure l'alignement rapide des processus métier ; les processus métier doivent au préalable avoir été intégrés horizontalement afin que l'information soit relayée entre les personnes de l'entreprise et les différents systèmes. IBM aime à dire que la SOA est appliquée jusqu'à sa propre organisation interne, c'est dire l'importance que ce concept revêt pour l'éditeur, présent depuis le début dans toutes les instances de spécification et de certification autour des Web Services. Contrairement à BEA, IBM n'implémente pas forcément dès leur ratification les nouvelles spécifications et préfère miser sur la maturité d'un noyau dur avant de se lancer.

Les produits

Historiquement, IBM est le premier à proposer un middleware orienté message, MQ Series, rebaptisé Websphere MQ. Début 2000, MQ Series Integrator (MQSI), basé sur MQ Series, est la première initiative EAI, très technique, de l'éditeur. Fin 2001, Big Blue s'offre CrossWorlds, ex-N°1 du marché de l'EAI, davantage tourné vers le métier et le processus. Cette acquisition marque le véritable coup d'envoi de l'offre EAI. La société ne s'interroge plus sur la concurrence entre serveur d'intégration et serveur d'applications : les deux ensembles seront complémentaires, regroupés au sein d'une même offre. Une nouvelle acquisition (Holosofx) apporte en octobre 2002 les briques de modélisation et de BAM ; si bien qu'en 2003, avec une offre rebaptisée **WebSphere Business Integration (WBI)**, IBM affirme, dans l'assemblage de ses produits, les complémentarités recherchées : entre EAI et Java, entre serveur d'intégration et serveur d'application, et enfin, entre métier et technique. Une progression de 82% des revenus 2002 de WBI récompense cette vision, sur laquelle se plaque l'approche SOA.

La tendance clairement affichée depuis début 2004, est la convergence des produits d'intégration et du serveur d'application : le serveur d'application devient la plate-forme d'exécution des briques d'intégration avec le produit WBI Server Foundation qui repose sur un socle WebSphere Application Server ; les connecteurs sont refondus sur une architecture JCA 1.5 devenue enfin bidirectionnelle, les collaborations deviennent des EJB, les processus sont décrits en BPEL et invoquent des Services Web réalisés dans le même atelier.

Ainsi, avec son serveur d'applications, son portail, et le rachat du produit Product Center de l'éditeur Trigo, dédié à la gestion des données de référence, IBM propose aujourd'hui une plate-

forme d'infrastructure complète, dédiée à la construction de SOA. La suite de développement s'intègre à celle du serveur d'application autour d'Eclipse et de la famille Websphere Studio, offrant ainsi un environnement unifié combinant les aspects portail, serveur d'application et intégration.

La figure ci-dessous replace chacun des composants dans la suite logicielle de l'éditeur :

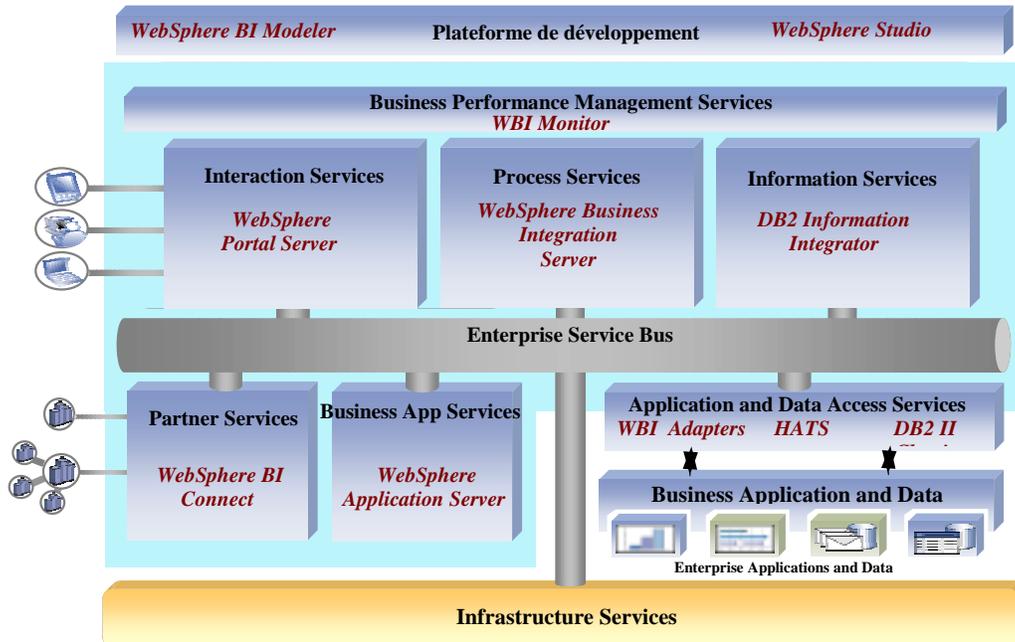


Figure 35 – Suite WebSphere

Dans ce schéma, l'ESB est le composant central qui assure l'intermédiation entre les couches de services et de données présentes dans toute l'entreprise, et un orchestrateur centralisé d'exécution et de supervision des processus.

Les briques du modèle SOA

De par la volonté ensembliste d'une APS et l'envergure d'un éditeur comme IBM, il n'est pas étonnant de retrouver ici couvertes et de façon performante toutes les briques du modèle.

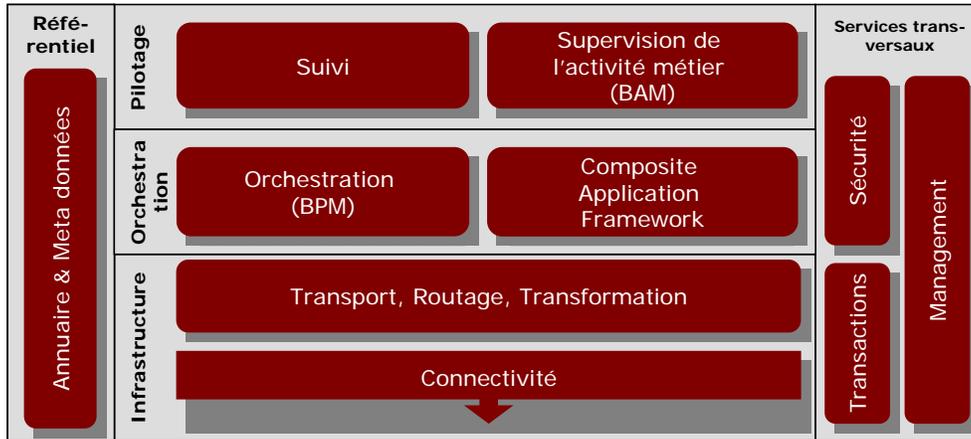


Figure 36 - Le modèle SOA de IBM

Les points forts de l'offre reposent sur les éléments proposés pour la gestion du processus métier. La version 5.1 de WBI propose un orchestrateur BPEL dénommé *Choreographer*, et en amont comme en aval, IBM fournit des outils de modélisation et de supervision des processus, facilitant les procédures d'accostage. A noter un bon niveau d'implémentation des standards Web Services et J2EE.

Intalio

Chaussee de Louvain 431H
1380 Lasne - Belgium
Tel: +32 (0) 2 357 82 36
eMail: europa@intalio.com

Intalio est une société américaine créée en 1999 avec un objectif : au service de l'accostage entre urbanisation et opérations, réaliser une jonction entre ateliers de modélisation de processus et les moyens de leur implémentation. Après 3 ans de R&D, la première mise en oeuvre chez un client eut lieu en 2001. Depuis, Intalio a développé des références significatives dans chacun des pays où la société est présente. En Angleterre, citons British Aerospace, Deutsche Post en Allemagne ou encore la Navy aux Etats-Unis. Aujourd'hui, l'offre d'Intalio est devenue la plateforme de référence pour General Motors dans le monde.

Conscient de l'implication de nombreux acteurs tiers à chacune des étapes et de la présence d'un existant en la matière dans les sociétés utilisatrices (nombreuses sont en France les sociétés déjà équipées d'un Aris ou d'un Mega, par exemple), il était capital qu'Intalio se montre moteur dans l'adoption et dans le respect des standards. La société est même allée bien plus loin puisqu'elle est à l'origine du site BPMI.org, promoteur du langage BPML, qui, comme nous l'avons vu ci-dessus, est globalement à l'origine du mouvement de normalisation qui s'opère à l'heure actuelle autour du processus métier.

Les produits

n³ Server joue un rôle intéressant dans cette volonté d'aligner le système d'information sur le métier et propose une passerelle entre le processus métier et le processus IT. Le produit se veut pragmatique : il s'agit concrètement d'outiller les démarches d'urbanisme et de cartographie en amont. Intalio propose l'instrumentation de ces démarches grâce à l'environnement n³ Designer et fait de son produit un moteur de BPM centralisé pour processus d'entreprise critiques et processus transversaux.

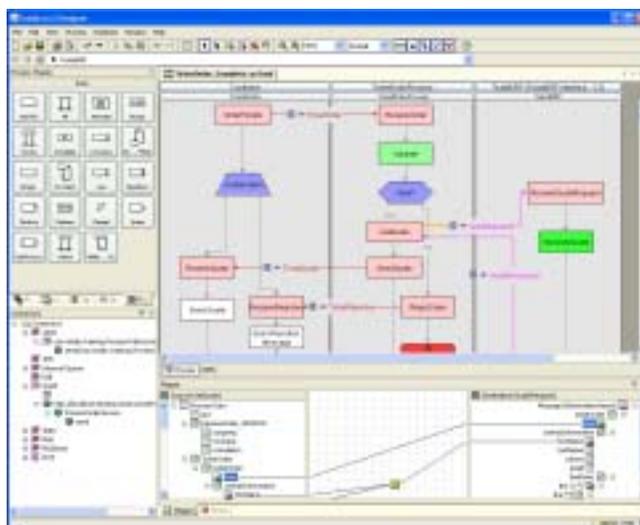


Figure 37 - n³ Designer

On retrouve sur cette figure différentes fonctions de l'environnement de développement, ici la représentation des processus au moyen de différentes *swimlanes* caractérisant les différents acteurs, métiers ou techniques, et, dans la partie basse de l'écran, on trouve un *mapping* entre deux formats de données.

En fournissant un lien naturel entre modélisation et implémentation des processus, Intalio cherche avant tout à réduire les coûts de développement, de maintenance et d'exécution dans leur mise en œuvre. L'idée phare est le découplage entre interface et implémentation, dont nous avons déjà affirmé l'importance pour la construction d'une SOA réellement urbanisée.

Le produit est autosuffisant mais propose cependant une passerelle avec l'allemand IDS Scheer, concepteur du logiciel de modélisation leader du marché, Aris. BPMN n'étant pas encore définitivement stabilisé (intégration dans le logiciel prévue courant 2005), les interfaces d'entrée-sortie avec Aris s'opèrent sur la base d'un langage propriétaire, AML (Aris Markup Language). Intalio opère alors en entrée une transformation d'AML en BPMN.

Les briques du modèle SOA de Intalio

Intalio couvre un modèle technico-fonctionnel relativement différent de celui des autres produits présentés dans ces pages. Cela s'explique par le fait que n³ est plutôt un outil de BPM permettant l'implémentation, l'exécution, le suivi et la supervision des processus. Ainsi, tout le cycle de vie du processus est couvert. Un processus pouvant être partiellement ou intégralement composé de services assemblés, n³ propose des connecteurs SOAP, et peut donc également être présenté comme un orchestrateur de services. Tout processus peut lui-même être exposé sous forme de service.

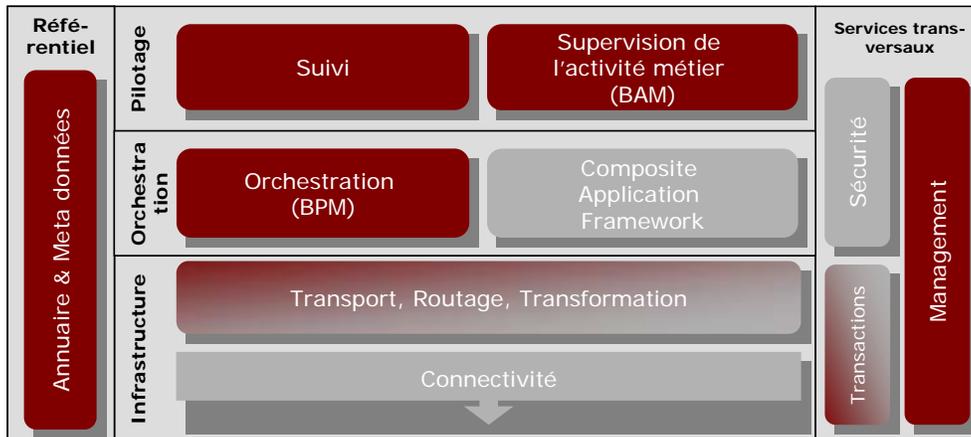


Figure 38 - Le modèle SOA d'Intalio

En effet, les couches basses ne sont finalement présentes qu'au niveau de quelques connecteurs et des outils de transformations. WebSphere MQ d'IBM est proposé en OEM. La grande majorité des connecteurs sont absents.

Le produit n'est pas spécialisé sur les Services Web et cela se vérifie dans l'absence de prise en charge des langages liés à la sécurité et aux transactions. A contrario, très orienté processus, le produit propose un orchestrateur complet, respectueux des standards, avec les outils d'administration, de suivi et de supervision nécessaires. Un annuaire natif à la structure granulaire organise l'ensemble, mais un partenariat avec Systinet existe également pour la couverture UDDI. n³ **Director** est la console de BAM et n³ **Console Administration** permet l'administration du système.

Microsoft

Service Clients Microsoft France
18 avenue du Québec
91957 Courtaboeuf Cedex

0 825 827 829

msfrance@microsoft.com

Bonne nouvelle : Microsoft, avant-hier tenant d'une approche fermée, est devenu hier apôtre de l'ouverture et de l'interopérabilité. BizTalk Server, serveur d'intégration EAI et d'orchestration SOA, existe depuis 1999. Plus significatif encore, les spécifications du protocole SOAP, ex XML-RPC, figuraient depuis cette date sur feu le site Web de BizTalk (biztalk.org). Aujourd'hui, Microsoft est fortement partie prenante, aux côtés de BEA et d'IBM, dans la définition des standards dédiés à la mise en œuvre des Web Services.

Au même titre que BEA par exemple, Microsoft a pris acte de la dimension architecturale et urbanistique des SOA et ne limite pas son travail à la fourniture de logiciels. La mutualisation des coûts, l'évolution par phase avec réutilisation des développements et l'intégration à la plate-forme font partie du discours et des préoccupations centrales de l'éditeur. Au-delà de nombreuses publications et articles fournissant des ressources illimitées sur son site Web, Microsoft annonce le lancement du projet Indigo, décrit ci-dessous.

Indigo

"Un framework unifié pour élaborer des applications orientées service sur la plate-forme Windows"

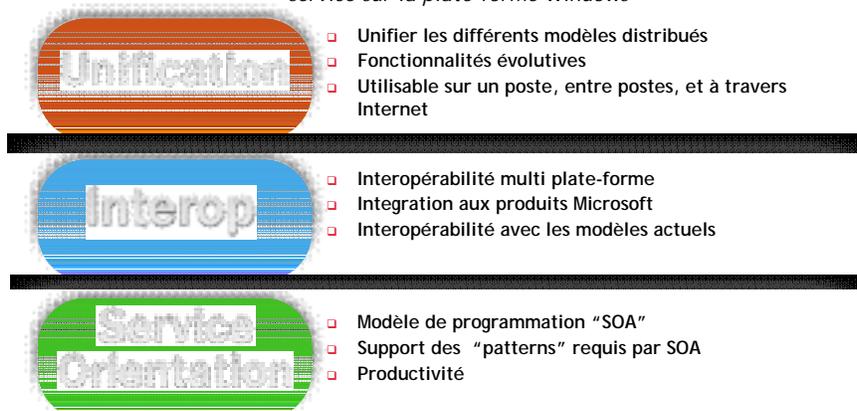


Figure 39 - Le projet Indigo de Microsoft

Les produits

.NET, l'architecture à partir de laquelle Microsoft a construit toute sa stratégie et toute son offre, préfigure l'avènement des APS.

BizTalk Server, ex-serveur EAI de l'offre de Microsoft, est devenu le composant central pour toute l'orchestration, le suivi et la supervision métier BAM. Ce positionnement confirme, s'il en était encore besoin, la convergence qui prévaut aujourd'hui entre intégration EAI et orchestration SOA. BizTalk assure le support de SOAP, WSDL, BPEL et UDDI. L'ensemble de l'offre repose sur XML et ses avatars (XSD, XSLT, BPEL...).

Les briques du modèle SOA de Microsoft

Fournisseur d'APS, Microsoft propose une large couverture du modèle.

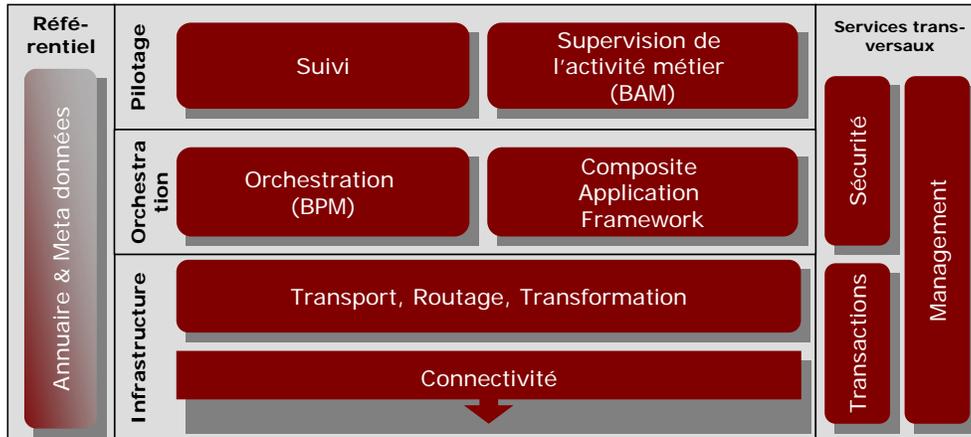


Figure 40 - Le modèle SOA de Microsoft

L'annuaire est Active Directory, non UDDI. Microsoft propose son propre middleware asynchrone, MSMQ. Hormis quelques connecteurs natifs (MQ Series, SAP), l'ensemble des adaptateurs provient du partenariat avec iWay, décidément omniprésent (Oracle, Sonic, SAP...).

Au niveau des services transversaux, la sécurité, les transactions et le management sont couverts, sans pour autant reposer sur des langages et protocoles propres aux services Web.

Ainsi, le support des transactions mainframes utilisera Transaction Integrator, et les transactions distribuées MS DTC ou les mécanismes XA fournis par Windows. En cela, l'intérêt est de faire jouer à plein l'effet plate-forme. Ajoutons également l'arrivée de Indigo qui apportera une couche de transactionnalité, respectueuse de WS-Coordination et incluse dans la prochaine version de BizTalk Server. Les transactions longues sont assurées par les mécanismes de compensation de l'orchestrateur BizTalk. La sécurité est assurée par la gestion de jetons de sécurité SAML et le Single Sign-On (SSO).

L'effet plate-forme joue également pour la brique de Management, puisque l'administration et la supervision technique du serveur d'orchestration sont assurées par une brique mutualisée, Microsoft Operations Manager.

L'orchestration de services et le suivi opérationnel reposent sur BizTalk Server. Visio peut être utilisé pour la représentation des processus de plus haut niveau : il propose une passerelle native bidirectionnelle avec BizTalk Server, et donc un bon moyen d'accostage au niveau processus. Le Composite Application Framework est couvert par l'ensemble des outils de développements de la plate-forme .NET.

La supervision de l'activité métier bénéficie de la dimension plate-forme de l'offre, puisqu'elle se compose de modules intégrés à Excel et à SQL Server.

Oblix

Dreve Richelle 161, Batiment N

1410 Waterloo, Belgium

+32 2 352 8772

oblix.benelux@oblix.com

Oblix est une société californienne créée en 1996. Elle a construit son offre sur la mise en relation des bonnes personnes avec les bonnes ressources, en fonction des droits attribués, et de permettre ainsi l'activation d'un business en ligne gérable et sécurisé : la sécurité est ainsi le cœur de métier d'Oblix et, selon IDC, l'un des marchés où vont se focaliser les investissements à très court terme. Sur son site Web, Oblix qui explique en quoi ses produits répondent à des enjeux sectoriels pour la finance (Sarbanes-Oxley, Bale II), la santé (HIPAA), le secteur public, le manufacturing et la logistique de transport.

La société a établi un ensemble de partenariats, notamment avec Microsoft et Tibco. Les produits sont disponibles sur plate-forme .NET et Unix (Solaris, Red Hat). Dans ses nombreuses références, on trouve notamment AXA Financial, i2 Technologies, Pfizer, Symantec et Volkswagen. Les investisseurs sont notamment les fonds d'investissement d'Intel, du Crédit Suisse, de Siemens ou encore J.P. Morgan.

Les produits



Oblix COREid (précédemment Oblix NetPoint) est une suite de modules dédiée à la gestion des identités. Le produit propose des fonctionnalités de SSO et de contrôle d'accès, de gestion des utilisateurs et des groupes, principalement pour des portails et des extranets. L'intégration avec les serveurs d'applications existants, les annuaires ou les plates-formes d'exploitation technique est couverte. Un module permet de dresser un *reporting* complet. Oblix SHAREid étend cette logique au monde du B2B. L'ensemble couvre ainsi l'intégralité du spectre de la sécurisation entre les applications et les services.



Le produit COREsv étend ce spectre aux interactions entre applications et entre services. Il fournit des outils graphiques pour construire des politiques de sécurité, les stocker et les distribuer à la demande, éventuellement par le biais de passerelles (gateways) qui interceptent les appels avant de les router. L'intérêt des passerelles est de réduire l'intrusion dans les applications et les services existants. Des agents dédiés à certains produits du marché (tels que Tibco Business Works) permettent la supervision de tous les appels entrants et sortants.

L'ensemble des activités réalisées par les passerelles et les agents peut être restitué sous format graphique avec COREsv Monitoring Dashboard. L'intérêt de ce composant est de permettre aux administrateurs de fixer les niveaux de SLA souhaités et d'être ainsi dynamiquement averti des franchissements de seuil, en temps réel.

Les briques du modèle SOA d'Oblix

La vocation d'Oblix est de se concentrer sur les fonctionnalités ayant trait à la sécurité. WS-Security, SAML et XML Signature sont ainsi présents dans l'offre. Quelques fonctionnalités de management viennent se superposer à cette couverture.

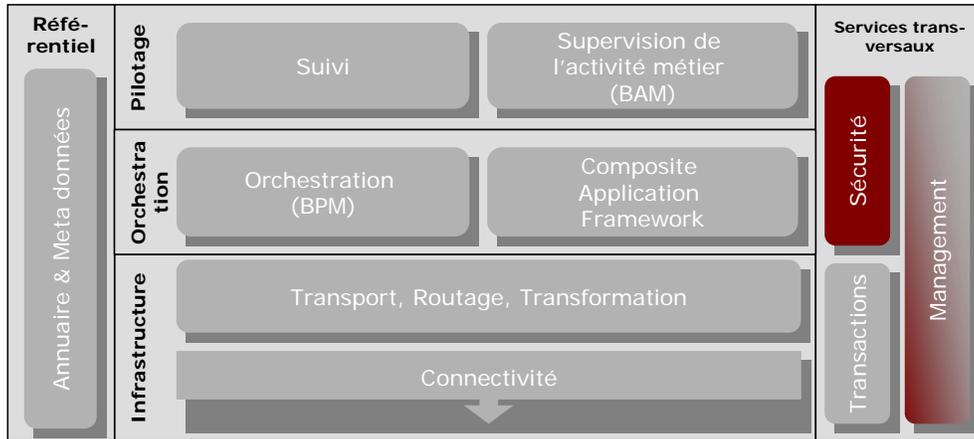


Figure 41 - le modèle SOA de Oblix

Oracle

Oracle France
15, boulevard Charles de Gaulle
92715 Colombes cedex
Tel: 01.57.60.20.20

En tant qu'éditeur de base de données et de serveur d'application, Oracle se positionne comme fournisseur de plate-forme d'infrastructure globale, autour de modules d'intégration et d'orchestration, de Business Intelligence, d'un portail d'entreprise, de la gestion de la mobilité, du RFID, d'un serveur d'application J2EE et de la gestion d'identité.

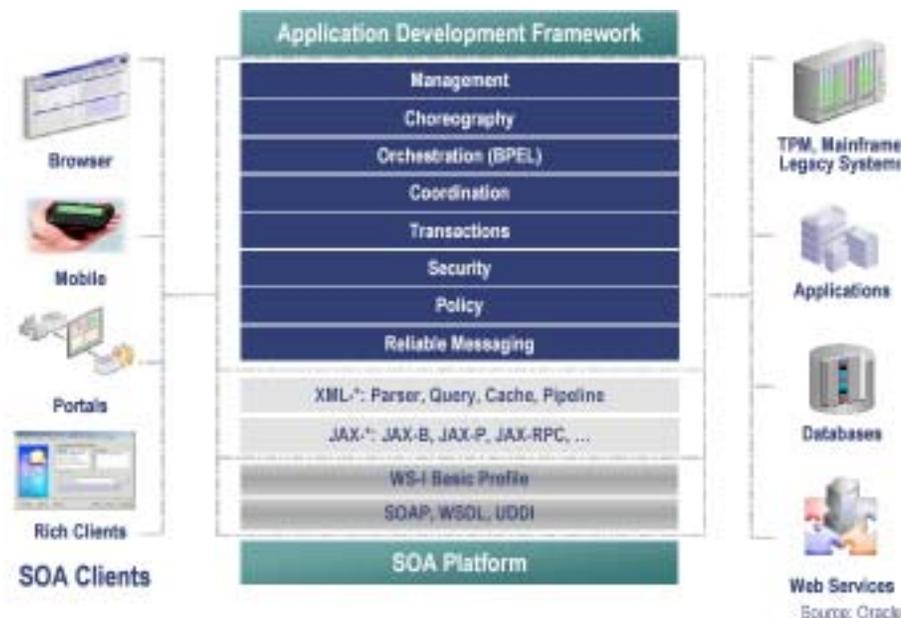


Figure 42 - La plate-forme APS d'Oracle

Oracle réaffirme également sa volonté d'implémenter les standards de façon large : en dehors des standards propres au monde des Services Web, on trouve ainsi RosettaNet, HL7, EDI, EDIFACT, AS2, UCCNet et SWIFT au niveau des bibliothèques et connecteurs applicatifs.

L'offre SOA bénéficie des éléments liés à la montée en charge et à la haute disponibilité de la plate-forme Oracle, et notamment au *grid computing*, avec la possibilité d'attribuer dynamiquement de nouvelles ressources en fonction des besoins.

Les produits

La plate-forme 10g propose un ensemble de modules qui positionnent Oracle parmi les APS. Outre les éléments déjà évoqués ci-dessus, citons la gestion des données de référence client assurée par Oracle Customer Data Hub.

La stratégie SOA d'Oracle se construit d'abord autour de BPEL Process Manager, produit qui affirme la prédominance de l'orchestrateur dans la vision de l'éditeur. BPEL Process Manager est composé de deux produits : le Server, environnement d'exécution, et le Designer, environnement de développement et de déploiement construit sur la plate-forme Eclipse. Le serveur est portable sur serveur d'applications BEA WebLogic.

Oracle affirme supporter des processus comprenant plus de 20 000 activités. La gestion des processus longs est prise en compte avec sauvegarde du contexte en base de données. Les interventions humaines peuvent être incluses dans le cours d'un processus au moyen d'une activité spécifique. Il est alors possible d'agir sur les données. Pour améliorer les performances, les messages au format XML sont encodés dans un format binaire géré par le système (BPEL Optimized SOAP Stack).

Ces produits sont issus du rachat de la société Collaxa, l'un des pure-players du domaine ; mais la volonté de l'éditeur ne s'arrête pas là et Oracle restructure l'ensemble de son offre afin de la mettre en conformité avec les attentes des entreprises en la matière. Ainsi, la base de données Oracle 10g va connaître d'ici à mi-2005 quelques enrichissements. Certains standards font leur apparition : WS-Reliability, WS-Security, WS-Policy

Les briques du modèle SOA d'Oracle

En tant que fournisseur d'APS, Oracle propose également un modèle complet.

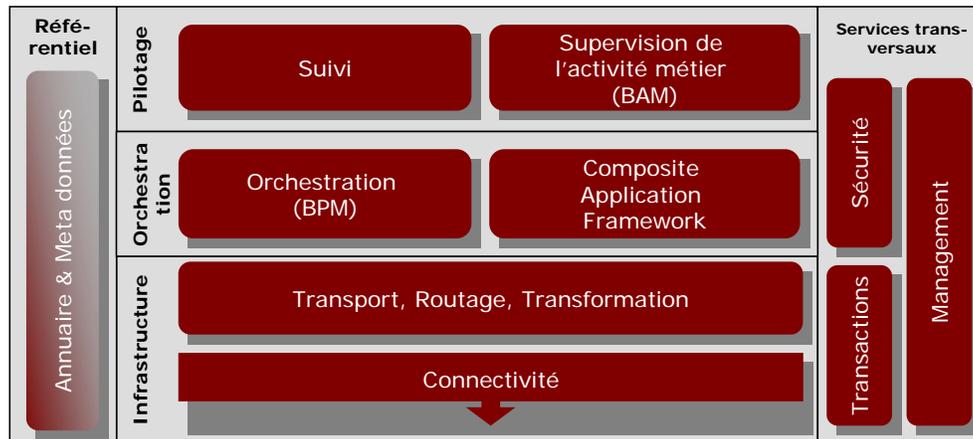


Figure 43 - Le modèle SOA d'Oracle

L'annuaire est tenu par Oracle Internet Directory (OID) qui n'est pas UDDI. La sécurité bénéficie d'un produit dédié, Oracle Identity Management. SAML et WS-Security sont implémentés au sein de OID.

En dehors des adaptateurs applicatifs déjà évoqués, un partenariat avec iWay permet à Oracle d'étendre ses capacités de connexion. Oracle possède son propre middleware asynchrone, Oracle Advanced Queue (AQ). La couverture des services transversaux est bonne : les mécanismes transactionnels sont délégués à l'orchestrateur et aux autres composants de l'offre (base de données, serveur d'applications).

La plate-forme de management des processus est le point fort de l'offre, avec une administration et un suivi complets de chaque instance de processus. 10g incorpore un produit de suivi temps réel de l'activité métier, dénommé BAM, *add-on* de la version 10.1.2 de l'Application Server, qui assure également la mise en corrélation des événements métier avec des éléments techniques, la comparaison d'événements unitaires avec des données agrégées stockées dans un *datawarehouse*, et, par conséquent, la constitution d'indicateurs clés de performance sur différentes échelles temporelles.

SAP

SAP France
59 bd Malesherbes
75008 Paris

01 55 30 20 00

Pas d'erreur : SAP mérite de figurer dans ce document. La firme de Darmstadt est devenue mondialement connue pour son progiciel de gestion intégré et ses modules fonctionnels (R/1 en 1972, R/2 en 1979, puis R/3 en 1992). La suite mySAP est ensuite venue enrichir le socle R/3 de modules dédiés à la gestion de la relation client (CRM), de la chaîne logistique (SCM), du capital humain (HCM), du cycle de vie des produits (PLM) et de la relation fournisseur (SRM). Enfin, depuis 2002/2003, SAP s'est lancé dans l'infrastructure, d'abord via son serveur d'applications WAS et son portail, puis, depuis 3 ans, via l'intégration d'applications avec XI (eXchange Infrastructure). La plate-forme APS Netweaver, offre d'infrastructure globale et fédérée, propose un aboutissement de cette logique. L'objectif est la réduction du coût total de possession, en rationalisant et organisant les voies de communication entre les modules fonctionnels et l'infrastructure transversale d'entreprise : une réelle logique d'urbanisme.

SAP annonce ainsi une *roadmap* officielle qui verra les modules fonctionnels de l'éditeur évoluer à l'horizon 2007 vers une architecture ESA (*Enterprise Service Architecture*). Aujourd'hui, certains modules sont encore en devenir et annoncés dans la roadmap : par exemple, l'annuaire de services, prévu pour la version de 2005. La philosophie de SAP se résume ici : permettre à ses clients de commencer à réaliser les premiers services d'infrastructure, la technologie stable et performante permettant de grandir et d'enrichir progressivement le socle au fur et à mesure que de nouvelles fonctionnalités arrivent avec les nouvelles versions. Cela signifie que la compatibilité ascendante est garantie.

Cette philosophie devrait convaincre de nombreux clients déjà équipés des modules fonctionnels de SAP. Les autres n'auront probablement pas tendance à faire immédiatement confiance à l'éditeur pour construire une telle plate-forme, a fortiori tant que les fonctionnalités d'infrastructure ne sont pas encore complètement au niveau des *pure-players* ou des autres acteurs du marché des APS. Cependant, faire le pari de SAP est une question qui mérite de se poser, l'éditeur n'ayant pas encore d'équivalent, de par sa double casquette fonctionnel/infrastructure et de par sa capacité à adresser un périmètre de grands comptes.

Les produits

Pour y parvenir, Netweaver propose un ensemble de modules dédiés à la gestion de l'infrastructure :

- On retrouve les modules incontournables déjà présents, avec le serveur d'applications WAS (Web Application Server), véritable fondation de Netweaver, pour développer des applications en langage ABAP ou en langage Java ; avec EP (Enterprise Portal), le portail d'entreprise ; et, bien entendu, avec XI, serveur d'intégration EAI.
- MI (Mobile Infrastructure), extension du portail et bouquet de technologies multi-canal ;

- Un outil de Knowledge Management, gestion de la connaissance et de l'information non structurée
- BI (Business Intelligence), un moteur d'analyse décisionnelle.
- MDM (Master Data Management), module de constitution et d'administration des référentiels de données d'entreprise
- Un moteur d'orchestration des processus métier

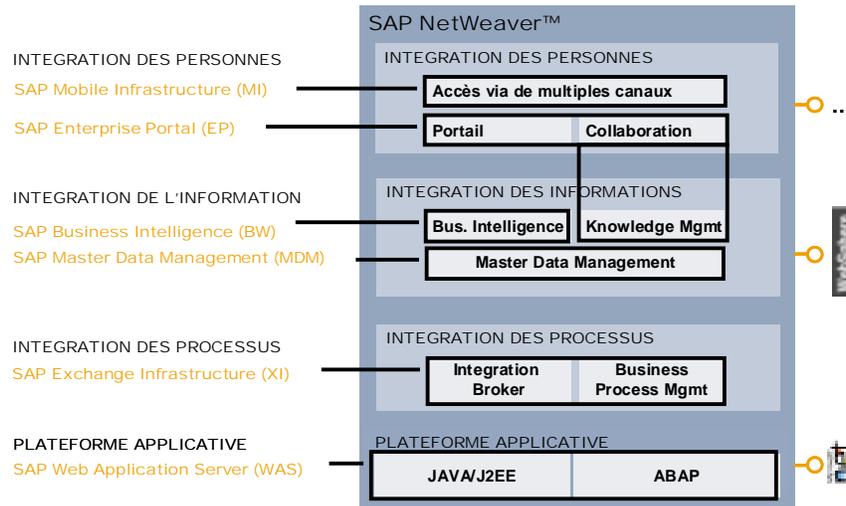


Figure 44 - L'offre SAP Netweaver

La philosophie de l'architecture est proche de celle des grands éditeurs du domaine, à savoir que chaque élément de l'offre est formé d'un ensemble de composants hébergés et exécutés par le WAS. C'est le cas de XI. C'est également le cas de modules tels que MDM, pour lequel la logique d'imbrication des modules est encore plus poussée, puisque MDM utilise le WAS, XI, BI et EP pour l'ensemble de ses fonctionnalités. Le tout est fourni avec environnement de développement et outils de supervision technique et fonctionnelle.

Les briques du modèle SOA de SAP

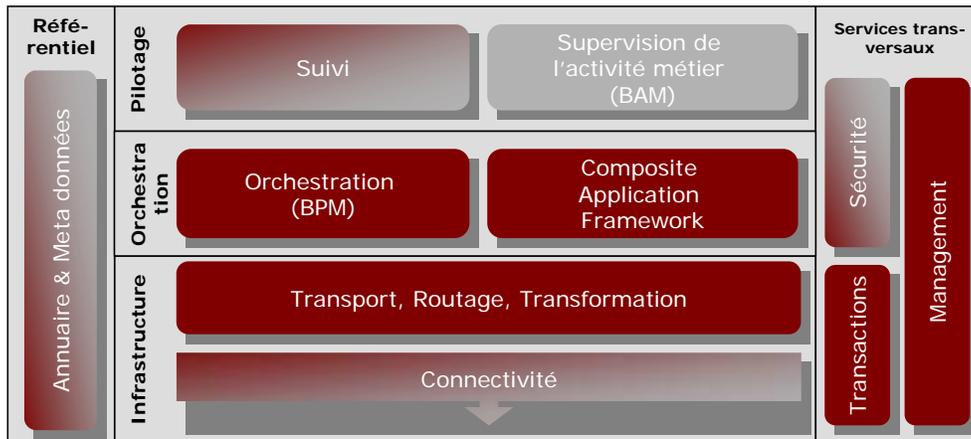


Figure 45 - Le modèle SOA de SAP

Ainsi que nous l'avons déjà évoqué, l'annuaire est annoncé pour la version 2005 de Netweaver. Concernant les couches basses, la connectivité est issue d'un partenariat avec iWay pour les aspects A2A et SeeBurger pour les échanges B2B. Les services transversaux sont assurés par des mécanismes internes natifs.

Le module d'orchestration est aujourd'hui partagé entre deux produits, l'un, interne à Netweaver, l'autre, assuré par Aris for Netweaver de IDS Scheer ; il sera à terme couvert par un produit unique, probablement celui d'IDS Scheer re-packagé pour la circonstance. Le lien entre tous les niveaux métier et technique sera alors complètement couvert. Le workflow demande une passerelle vers SAP Workflow.

Le suivi des flux est assuré par une console interne, alors que le suivi des processus demande encore une bascule vers les modules fonctionnels. La supervision métier BAM est absente.

SeeBeyond

SeeBeyond France

7 place d'Iena

75116 Paris

téléphone : +33 (0) 1 53 57 58 58

fax : +33 (0) 1 53 57 57 77

info_france@SeeBeyond.com

En 1991, Jim Demetriades crée la société STC (Software Technology Corporation), marquant l'avènement d'une des plus anciennes plates-formes d'intégration : Data Gate. STC s'implante en France à la fin des années 90 et signe quelques belles références parmi lesquelles Axa, la Fnac ou encore Leroy Merlin. Au tournant du millénaire, la société et le produit changent de nom, pour respectivement devenir SeeBeyond et e*Gate. Enfin arrive en 2003 la nouvelle version de ce qu'il est désormais convenu d'appeler une suite : ICAN (Integrated Composite Application Network), cinquième génération de l'offre.

130 millions de dollars furent investis en 3 ans pour réussir cette mutation d'une suite qui s'est ainsi déplacée du marché de l'EAI vers celui des plates-formes applicatives APS, positionnement confirmé très tôt par le Gartner Group dans ses cadrans magiques, ce que confirme également le choix du nom ICAN, produit dédié à l'assemblage d'applications composites. Le groupe Carrefour et la SNCF ont fait le choix de cette version très respectueuse des standards du monde J2EE et des Services Web.

Les produits

En plus de ICAN, SeeBeyond propose SeeBeyond ESB, potentiel ticket d'entrée à son offre, qui regroupe les fonctionnalités suivantes :



Figure 46 - Modèle technique de SeeBeyond ESB

Pour SeeBeyond, il existe une valeur ajoutée à être vendeur d'EAI sur le segment de l'ESB : c'est la capacité d'intégration de ce module au reste de son offre. Ainsi, les connecteurs applicatifs ne sont pas ceux de sociétés tierces, et partagent le référentiel de la plate-forme pour mutualiser les composants et privilégier la réutilisabilité. De plus, eInsight propose nativement la gestion des interventions humaines dans le cours du processus et des affectations de tâches, avec le module Worklist Manager. Enfin, l'offre sait s'interfacer avec les différentes technologies d'annuaire.

SeeBeyond ESB indique bien le périmètre attendu d'un tel produit :

- Un *backbone* de *queuing* JMS,
- Les connecteurs fichier, base de données et Web Services,
- L'environnement de développement,
- Le versioning des services et processus,
- La gestion des environnements et du déploiement,
- L'annuaire de services et une connectivité LDAP,
- Un monitoring JMX,
- L'utilisation de *elnsight* assure un développement essentiellement basé sur du paramétrage, sans rédaction de code Java (mais les composants se déploient sous la forme d'un EAR contenant des EJB).

De plus, il offre, autour des environnements de développement, d'exécution et de supervision, certaines fonctionnalités d'accostage, telles que l'analyse d'impact et la gestion avancée des versions de services et de processus, sont présentes.

La génération de Services Web par empaquetage de bases de données ou d'API existantes est présente. Cela permet, lors de la conception des processus, de définir chaque activité comme un appel de service. L'ensemble du processus est défini en BPEL.

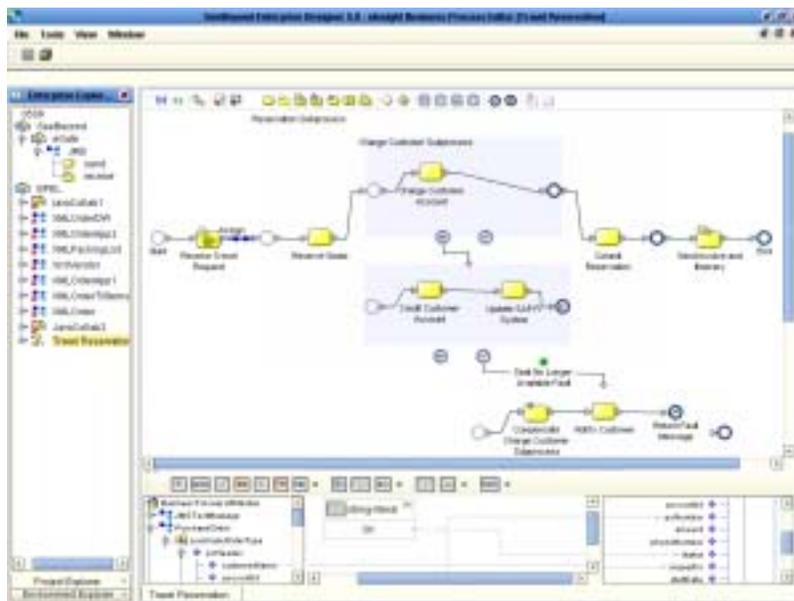


Figure 47 – Représentation d'un processus elnsight

Sur le socle ESB, ICAN apporte des connecteurs supplémentaires, la capacité à développer en langage Java, un portail, un outil de gestion des références croisées, et tout l'atelier de supervision BAM. Chacun de ces modules peut être ajouté progressivement, de façon évolutive, sur le socle existant.

SeeBeyond ESB et ICAN peuvent être tous deux considérés comme des orchestrateurs SOA, l'un étant limité à l'orchestration et à l'intégration, l'autre s'élargissant à fournir une couverture complète d'infrastructure.

Les briques du modèle SOA SeeBeyond

La couverture du modèle SOA par SeeBeyond est assez large et se rapproche de celle d'une APS, témoignant de la mue entamée par cet éditeur.

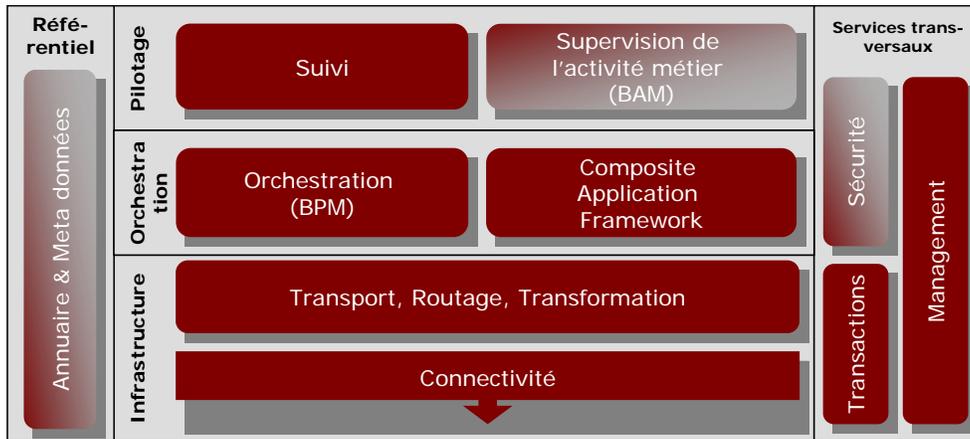


Figure 48 - Le modèle SOA de SeeBeyond

Aucune brique n'est absente, mais certaines sont plus ou moins bien couvertes. Les fonctions d'annuaire, notamment, sont limitées par le fait que l'annuaire, bien qu'au format UDDI, reste empaqueté au service unique du serveur ICAN. D'autre part, les fonctionnalités de BAM semblent encore relativement limitées face à ce qu'offrent aujourd'hui certains challengers présents depuis plus longtemps sur ce segment de marché. Pour le reste, l'ensemble des fonctions est couvert avec un bon niveau de service, et l'offre s'est enrichie de la brique Composite Application Framework, présentant un environnement de développement de composants Java et un portail.

Sonic Software

Sonic Software S.A.R.L.

3 Place de Saverne

Les Renardières B

92901 Paris la Défense

Tel: +33 1 41 16 1604

Fax: +33 1 41 16 1601

Email: info-fr@sonicsoftware.com

Sonic Software est une filiale de Progress Software, dédiée à fournir des produits et des services pour l'entreprise temps réel. Son produit phare, Sonic ESB, permet l'instrumentation d'architectures orientées Services.

Les produits

S'il est un produit qui incarne clairement la notion d'ESB sur le marché, dans la définition qu'en donne le Gartner Group et au travers des compléments que nous avons souhaités apporter précédemment, c'est définitivement Sonic ESB, dont la figure 32 représente la structure d'offre.

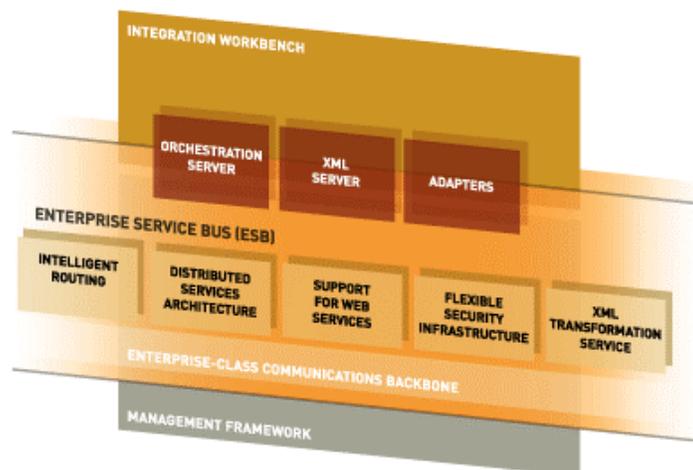


Figure 49 - L'offre ESB de Sonic Software

Le produit repose sur un bus asynchrone distribué, Sonic MQ, reconnu pour ses excellentes performances, et qui propose les fonctionnalités déjà évoquées de routage dynamique (Content Based Routing), des fonctions de transformation, le support des Web Services, le support natif du transport XML, la sécurité d'accès au bus et aux *endpoints* et une interface de supervision centralisée. Trois produits viennent enrichir les fonctionnalités de bus :

- Sonic Orchestration Server** est l'orchestrateur, qui étend les fonctionnalités de base du bus pour y ajouter le support des transactions longues et des flux complexes de données, via la modélisation des processus, leur exécution et leur supervision.
- Business Process Manager** complète le bus par les fonctions de BPM.

- ❑ **Sonic XML Server** est un outil de traitement optimisé (parsing, agrégation, transformation XSLT...), de stockage et de requêtage XQuery des documents XML, qui vient se brancher sur le bus ESB.
- ❑ **Integration Workbench** est l'environnement de développement unifié pour l'ensemble des serveurs d'exécution de la suite.

Les briques du modèle SOA de Sonic

La couverture du modèle SOA par l'offre de Sonic Software est caractéristique de celle qu'un ESB est susceptible de fournir :

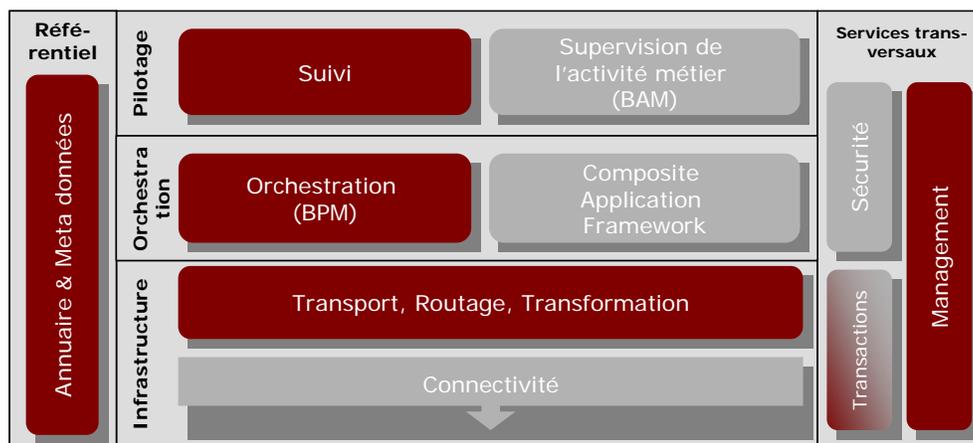


Figure 50 – Le modèle SOA de Sonic Software

Les services d'annuaire sont internes au produit et permettent la description des services et des méta-données décrits dans l'environnement de développement. Au niveau de l'infrastructure, Sonic Software dispose d'un middleware asynchrone performant et réputé. L'offre ne propose pas de connecteurs natifs mais se repose sur des partenaires tiers comme iWay. Les fonctions d'orchestration, de suivi et d'administration sont présentes. Dans les services transversaux, la gestion des transactions est couverte par des mécanismes natifs, mais la gestion de la sécurité fait encore défaut. Parmi les services d'orchestration, le Composite Application Framework est manquant, car ce n'est pas le périmètre du produit (les fonctions de workflow sont néanmoins gérées nativement). Au niveau du pilotage, on trouve les outils de suivi d'exécution mais pas d'outils de BAM de plus haut niveau.

Sterling Commerce

Tour Franklin - 26e Etage

100-101, Terrasse Boieldieu

La Defense 8

92042 Paris La Defense Cedex – France

+33.1.55.23.60.00

Créée il y a 30 ans et présente depuis de nombreuses années dans le domaine de l'échange EDI/B2B, puis, plus récemment, comme fournisseur d'une plate-forme d'intégration (Sterling Integrator), Sterling Commerce poursuit la logique de son positionnement (35% de parts de marché sur l'EDI et 48% sur les transferts sécurisés de fichiers) et a annoncé le 15 novembre 2004 sa nouvelle stratégie : la Multi-Enterprise Collaboration.

L'optimisation des processus liés à la chaîne d'approvisionnement représenterait un vecteur de croissance pour bon nombre d'entreprises. Pour le maîtriser, Sterling Commerce propose d'offrir une visibilité globale sur l'activité d'une entreprise en y incluant les partenaires. Cette stratégie est soutenue par une plate-forme technologique dédiée, MESA (Multi-Enterprise Services Architecture), qui étend l'apport des SOA aux partenaires de l'entreprise, et notamment les capacités d'évolution et d'alignement que ce type d'architecture peut proposer. L'orientation Processus autour des Services Web est fortement marquée.

D'autre part, Sterling Commerce, sensible aux enjeux liés à la sécurité concernant ce type d'échanges, a signé avec Entrust un partenariat destiné à fournir des fonctionnalités de gestion d'identité et de signature unique. Les premiers produits devraient apparaître début 2005, reposant sur un socle transversal de sécurité nommé Secure Commerce Foundation. Des déclinaisons verticales (distribution, finance, industrie, santé) de l'offre sont annoncées.

Les produits

Sterling Integrator est le logiciel dédié à l'exécution et l'exploitation d'échanges de données A2A/B2B, cœur du positionnement EAI/SOA.

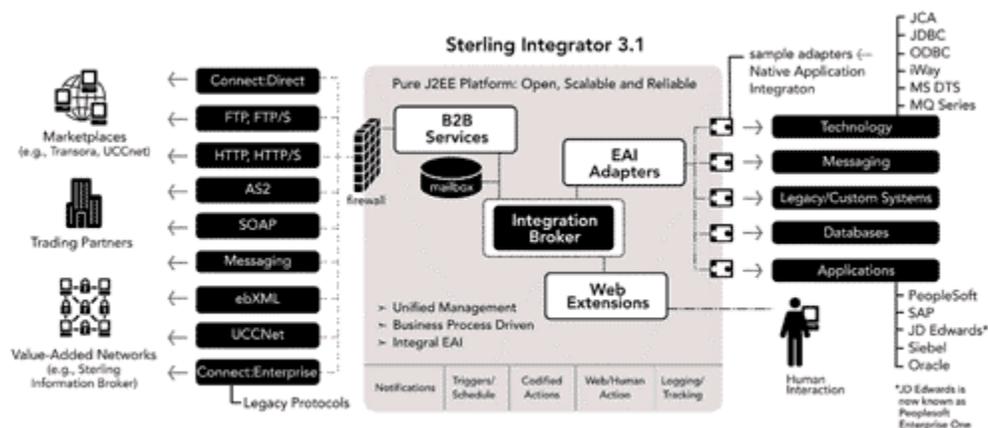


Figure 51 - Sterling Integrator

Le produit Gentran est un intégrateur EDI traditionnel capable de gérer des formats EDI et des langages XML. Il utilise le même mapper que Sterling Integrator. Spécialiste de l'échange B2B, Sterling Commerce propose des bibliothèques UCCNet, RosettaNet, ebXML, d'AS1 et AS2.

D'autre part, Sterling Commerce propose Connect, une solution dédiée à l'exécution et l'exploitation des échanges sécurisés de fichiers. Connect/Express est notamment spécialisé dans l'échange de fichiers de type CFT, XFB et Inter.Pel.

Au-delà de ses produits, Sterling Commerce est également fournisseur de réseau d'échanges B2B à valeur ajoutée via Sterling Information Broker, récemment certifié conforme SAS 70 Type II, soit conforme au Sarbanes Oxley Act.

Les briques du modèle SOA

MESA est construit sur quatre ensembles technologiques :

- Des services d'intégration, regroupant les couches basses du modèle SOA ;
- Des services d'orchestration de processus incluant les interventions humaines ;
- Des services de Business Intelligence qui assurent les fonctionnalités identifiées dans la brique de supervision métier ;
- Des services de traitement et de présentation de l'information, qui assurent les fonctionnalités identifiées dans la brique Composite Application Framework.

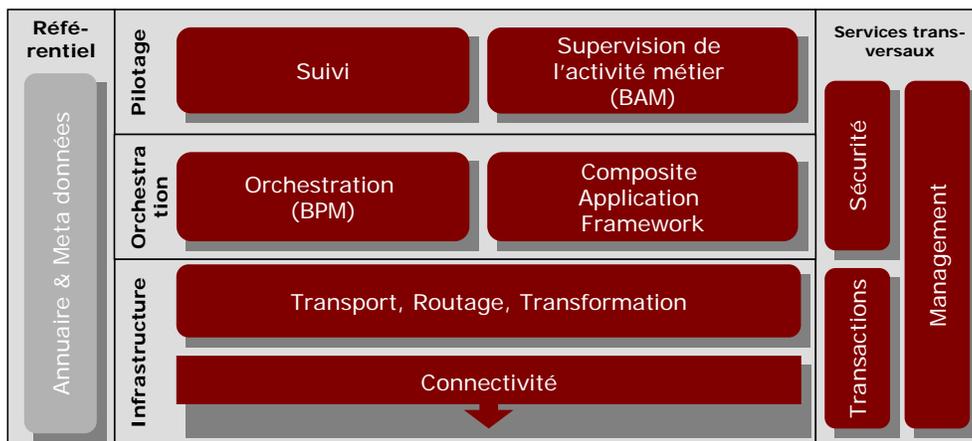


Figure 52 - Le modèle SOA de Sterling Commerce

La gestion de contexte fait partie intégrante de la vision. Concernant la sécurité, grâce au partenariat avec Entrust, on trouvera dans l'offre des fonctionnalités de chiffrement et de gestion des certificats, de gestion des identités, d'application des politiques et de Single Sign-On.

Systinet

Systinet France SAS

23, Rue Balzac

Paris, 75008, France

Phone: +33 (1) 53 53 67 28

Fax: +33 (1) 53 53 68 29

Email: sales-france@systinet.com

Systinet est à l'heure actuelle un des pure-players majeurs du monde des SOA. La compagnie est principalement implantée aux Etats-Unis et en Europe, avec des bureaux en France, aux Pays-Bas et en République Tchèque. Son siège est aux Etats-Unis et sa R&D, basée à Prague, compte environ quatre-vingt collaborateurs. La société compte environ 150 collaborateurs dans le monde.

Les premiers produits furent commercialisés en août 2000 et ont été plusieurs fois primés, notamment par le Business Integration Journal (ex-EAI Journal) en tant que meilleure solution de Web Services en 2003, et par la Software & Information Industry Association en 2003 et 2004, dans la même catégorie de produit.

La politique de la société est d'implémenter les standards stabilisés, dans les moindres détails, au plus près de la spécification, sans extensions propriétaires qui limitent les possibilités de portage et d'évolution. Le priorité est clairement placée sur l'interopérabilité : Systinet est par ailleurs membre actif du W3C, d'OASIS et de WS-I.

Plus de 150 sociétés parmi le Global 2000 utilisent aujourd'hui les produits de Systinet. Une des plus belles références est probablement Amazon, précisément le cas cité en introduction de ce livre blanc.

Parmi les partenaires de Systinet également mentionnés dans ce document figurent Tibco et Intalio, auxquels Systinet fournit notamment sa technologie d'annuaire dans la perspective de construction des SOA. La société se présente donc comme technologie complémentaire permettant à ces éditeurs de positionner leurs offres (respectivement EAI et BPM) sur des architectures orientées service.

Les produits

L'offre de Systinet se compose de plusieurs briques :

- **Systinet Server 5.5** agit comme un container Java/C++ et permet de mener toutes les étapes de développement : mise au point, tests et déploiement, depuis Eclipse par exemple. Les fonctionnalités de Systinet Server couvrent notamment la génération automatique de fichiers WSDL à partir d'un fichier *.class* Java, sans recompilation de la classe et la génération, à partir de l'interprétation d'une définition WSDL d'un service, du *stub* d'appel des services d'un client externe. Le serveur se déploie en *standalone* ou sur tout serveur J2EE, offrant l'interopérabilité entre différentes versions de ces serveurs ainsi qu'avec .NET, et supporte les principales plates-formes C++. SOAP 1.2 (sur HTTP 2.0) et WS-Security sont implémentés.

- **Systinet Server for MOM Services 5.5** (précédemment Systinet Gateway), offre, comme son nom l'indique, une passerelle pour les principaux *middlewares* du marché. Selon le sens par lequel on l'aborde, Systinet Server for MOM Services représente autant l'apport d'un côté asynchrone au modèle des Web Services que l'abstraction sous forme de Web Services des couches MOM traditionnelles, tout en masquant les aspects propriétaires de cette connectivité. Les serveurs sont disponibles pour Tibco RendezVous, MQ Series, Sonic MQ (connexion native) et pour tout bus accessible par l'interface JMS. Systinet Server for MOM Services supporte les principaux protocoles de sécurité, au niveau de la couche de communication (HTTPS, HTTP basic, HTTP digest, X.509) comme de la couche Web Services (XML Encryption, XML Signature, WS-Security). A cela s'ajoute le support de WS-Addressing et WS-ReliableMessaging.

Enfin, la solution Systinet Registry constitue le cœur de l'offre de Systinet. Sa version courante, la 5.5, est implémentée sur la spécification UDDI V3 de l'OASIS. L'utilisation de cet annuaire est autant applicable à un contexte interne qu'à un contexte externe. La solution propose des fonctionnalités de réplication entre plusieurs instances d'annuaire. Son objectif principal est de gérer le cycle de vie des services et de fournir des fonctionnalités avancées de gestion des méta-données de l'entreprise.

Les briques du modèle SOA de Systinet

Au travers de ses trois produits, Systinet couvre le périmètre suivant du modèle SOA.

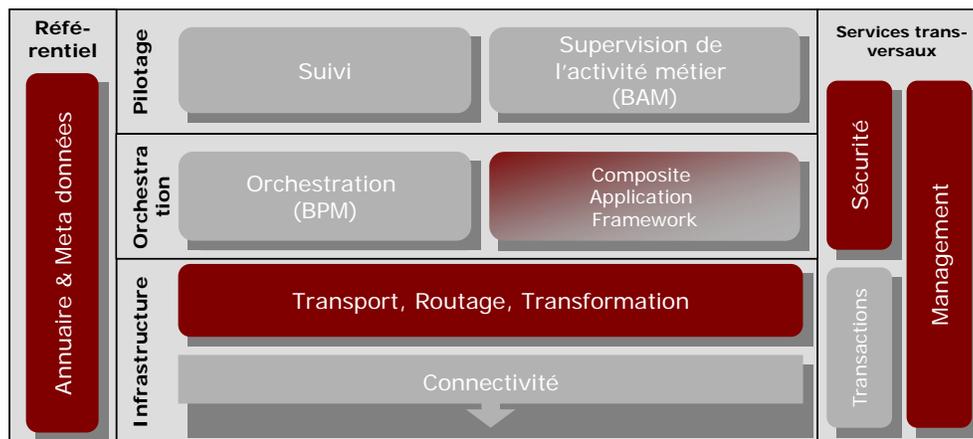


Figure 53 - Couverture du modèle SOA par Systinet

L'ensemble des fonctionnalités proposées est regroupé par l'éditeur dans le schéma suivant :

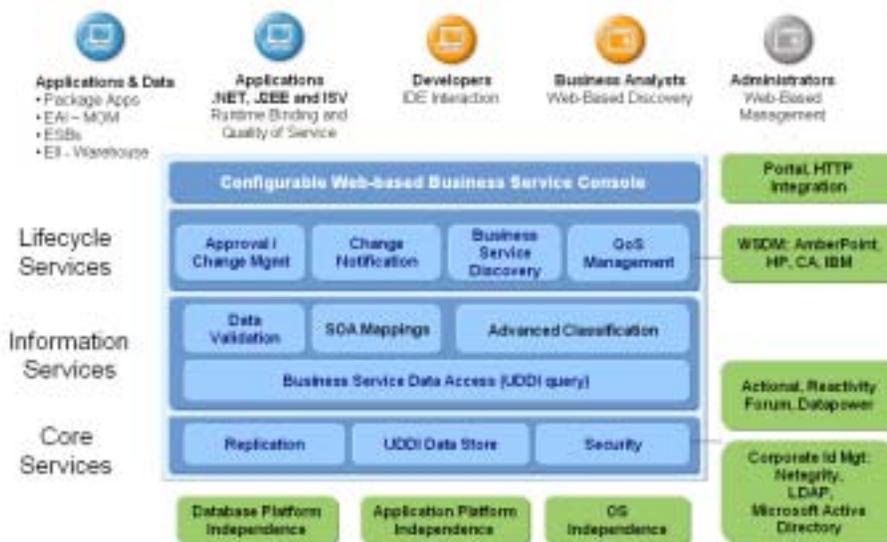


Figure 54 - Couverture fonctionnelle et technique Systinet

Au-delà des qualités technologiques de sa solution et des fonctionnalités qu'elle apporte, Systinet met l'accent sur la nécessité de définir au préalable les taxonomies permettant de catégoriser, nomenclaturer et classifier les différents services ; l'annuaire se charge ensuite de la gestion des taxonomies sur un plan opérationnel, y compris dans le requêtage pour l'accès aux services. Un navigateur permet alors de parcourir le contenu de l'annuaire selon des critères de recherche basés sur ces catégories.

La gestion dynamique du cycle de vie des services est également prise en charge : procédures d'approbation des services, notification des changements, gestion de la qualité de service. L'annuaire offre ainsi le niveau d'indirection qui garantit un couplage lâche des Web Services et représente pour Systinet la fondation d'une SOA.

Tibco

4, Place de la Défense
La Défense 4
92974 Paris la Défense Cedex
France
Phone: +33 1 58 58 28 02

Tibco est l'un des acteurs majeurs, depuis plusieurs années maintenant, de l'intégration EAI, et souvent présenté comme le leader mondial, se partageant au gré des classements et des années cette place avec IBM. Curieusement, l'offre de Tibco est à la fois un candidat parfait et un élève buissonnier à la mise en œuvre d'une SOA :

- un candidat parfait, car Business Works est depuis sa version 5 suffisamment performant pour s'affranchir du contexte EAI strict et être présenté comme un véritable orchestrateur de services ;
- un élève buissonnier, car le respect de certains standards tarde encore à s'affirmer dans les produits de l'éditeur, ce que Tibco explique par la difficulté d'identifier le langage, entre BPSS et BPEL, qui s'imposera demain. L'éditeur, étant membre du groupe de spécification et certification BPEL, annonce néanmoins dans sa *roadmap* le support de ce langage pour le début de l'année 2005.

Les produits

Le besoin d'organiser les services, d'obtenir des références communes sur les données (donc de rationaliser pour mieux piloter), les problèmes de coûts et de redondance, ont été perçus très tôt, orientant l'offre vers une meilleure réutilisabilité et une mutualisation des composants de toute granularité. Ainsi, en dehors de Business Works, l'orchestrateur, Tibco dispose d'un module incontournable, XML-Canon, qui permet l'import/export de tout composant du référentiel opérationnel d'exécution au format XML.



Figure 55 - Positionnement de XML-Canon dans l'offre de Tibco

L'ensemble des *assets* peut être stocké dans le référentiel XML-Canon, et notamment les XML-Schemas et DTD pour la description des méta-données. Si XML-Canon n'est pas à proprement parler un environnement de modélisation, il n'en dispose pas moins de fonctions de gestion de

référentiel (versioning, analyse d'impact...) disposant d'un accostage natif avec le serveur d'exécution. Les méta-données peuvent être catégorisées et des attributs spécifiques et personnalisés peuvent leur être attribués.

XML-Canon assure ainsi l'organisation des données, des services et des processus, en attribuant des responsabilités (sécurité via listes de contrôle d'accès, procédures d'approbation suite à toute modification) sur chacun des éléments du système. La définition des Services Web peut être administrée depuis ce produit, même si celui-ci ne peut prétendre jouer le rôle d'annuaire. Ce rôle sera dévolu, par partenariat, au produit Registry de l'éditeur Systinet.

Une vraie SOA demande une architecture de messaging solide : c'est précisément ce que propose RendezVous, le bus de messagerie asynchrone. Tibco parle ainsi de *real-time event-driven architecture*. L'éditeur met l'accent sur la possibilité de faire du *late-binding*, ce qui signifie séparer le processus du transport pour les rendre complètement indépendants et découpler ainsi l'infrastructure de l'orchestration. Enfin, tout processus peut être exposé sous forme de service.

L'environnement de développement Business Works est unifié pour l'ensemble de l'offre.

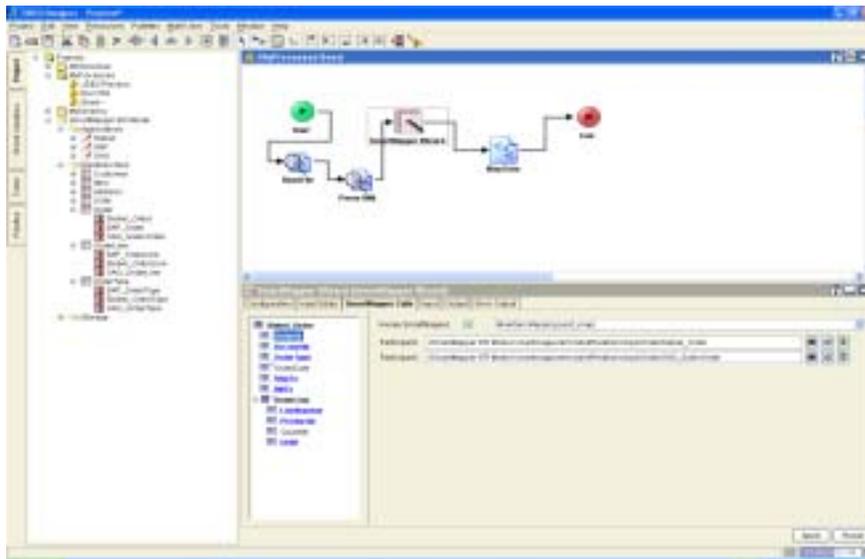


Figure 56 - L'environnement Business Works

Les briques du modèle SOA de Tibco

Le modèle SOA de Tibco est assez complet, mais il trouve ses limites dans l'implémentation de BPEL et de certains standards liés à la sécurité des Web Services, aux transactions et à la sécurité, limites que les partenariats avec Systinet et Oblix tendent à gommer.

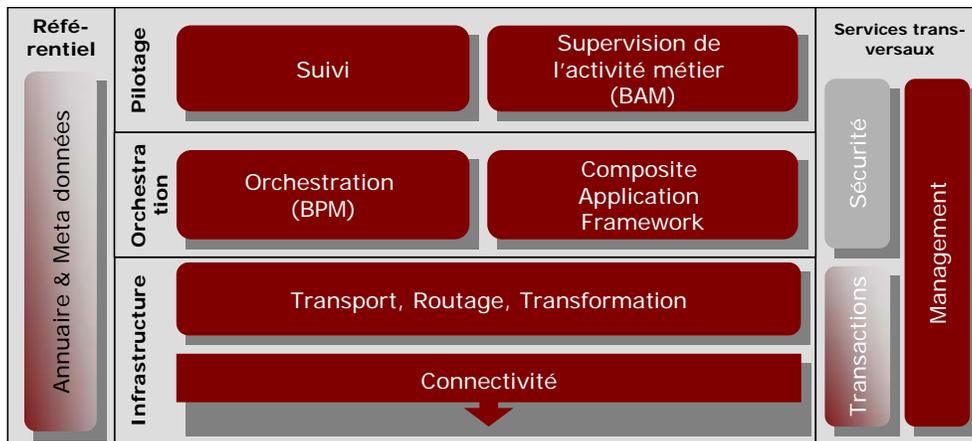


Figure 57 - Le modèle SOA de Tibco

Les couches basses font depuis plusieurs années maintenant partie intégrante du modèle EAI/B2B de l'éditeur. Le modèle de sécurité des Services Web est absent : la sécurité dans l'offre de Tibco repose sur les mécanismes natifs offerts. Les produits de Tibco proposent ainsi des options de sécurité, telles que la gestion d'ACL assez fines ou la connexion LDAP.

Concernant l'orchestration et le Composite Application Framework, Tibco propose depuis plusieurs années maintenant des produits crédibles et complets en terme de fonctionnalités. Au niveau de la supervision métier et du BAM, les produits de Tibco sont très avancés, probablement les plus complets du marché.

webMethods

webMethods France
58 Avenue Hoche
75008 Paris
Tel: +33 (0) 1 55 37 36 00

Acteur phare du marché de l'EAI depuis de nombreuses années, leader français du marché des EAI d'entreprise, webMethods a entrepris sa mue vers l'APS en opérant une batterie de rachats à la fin de l'année 2003 :

- Le rachat de Dante Software lui permet de se doter, avec le produit Optimize, d'un outil de BAM ;
- Celui de Data Channel apporte à l'offre une brique de portail d'entreprise ;
- Enfin, l'acquisition de The Mind Electric apporte la capacité à construire des architectures orientées Services à partir des produits Fabric et Glue.

Les produits

A partir de sa couverture du périmètre EAI, webMethods est réellement crédible sur le marché de la SOA :

- **Glue** assure l'infrastructure de création et d'invocation de services ;
- **Fabric** gère pour le référencement et l'assemblage de services ;
- **Modeler**, composant intégré à l'offre d'EAI, orchestre les processus.

Le produit Glue de The Mind Electric a rapidement été adopté par une importante communauté d'utilisateurs à travers le monde. L'objectif principal du produit est d'encapsuler sous forme de service Web tout composant, qu'il soit implémenté en J2EE comme en .Net.

Ses caractéristiques techniques sont :

- Sa capacité à publier un objet sous forme de Web Services par le jeu d'une instruction unique ajoutée au code du composant.
- Des niveaux de performance de l'ordre de 1000 messages à la seconde, pour une empreinte mémoire très faible, de l'ordre de 600 Ko.
- Le support des différents types de messages SOAP (RPC/Encoding, Document/Literal et RPC/Literal) et des attachements.
- Support de plusieurs mécanismes de sécurité : SSL, HTTPs, JAAS, WS-Security, certificats, annuaires...
- Déploiement à distance et à chaud.

Avec Glue, on met en place l'infrastructure d'accès aux services en les encapsulant dans des protocoles standardisés répondant aux besoins de neutralité technologique et de séparation interface/implémentation. Avec Fabric, on positionne le tiers d'intermédiation et on construit l'infrastructure qui assurera le découplage entre les services. Plutôt que de se connaître les uns les autres et de s'auto-référencer « en dur », les producteurs de service vont déléguer leur référencement à un tiers qui assurera aux consommateurs l'accès à leur interface. Fabric

constitue ce tiers de confiance, ce *tissu* (la traduction française du faux ami *fabric*) qui assure un couplage lâche entre producteurs et consommateurs de services.

Les caractéristiques techniques de Fabric sont :

- **Annuaire de services** : référencement et mise à disposition des interfaces pour une découverte dynamique par les consommateurs de service ;
- **Haute performance et disponibilité** : l'architecture est constituée d'un ensemble de nœuds qui permettent la distribution de la charge. Un nœud peut également compenser une rupture dans l'accès aux services ;
- **Architecture 100% peer-to-peer** : au niveau de chaque nœud est distribuée une instance de l'annuaire de services.

Principale limitation du modèle : tout composant/service qui souhaite être répertorié dans Fabric doit voir son code modifié pour faire appel à l'API Fabric. S'il ne s'agit que d'une simple ligne, il n'en reste pas moins que le contexte est intrusif et qu'il peut s'avérer fastidieux, voire pénible à maintenir, pour les configurations importantes.

Les briques du modèle SOA de webMethods

Grâce à ses acquisitions, webMethods propose un modèle complet qui doit encore cependant faire ses preuves en matière d'intégration des différentes briques entre elles, ce que prévoit cependant la *roadmap* de l'éditeur.

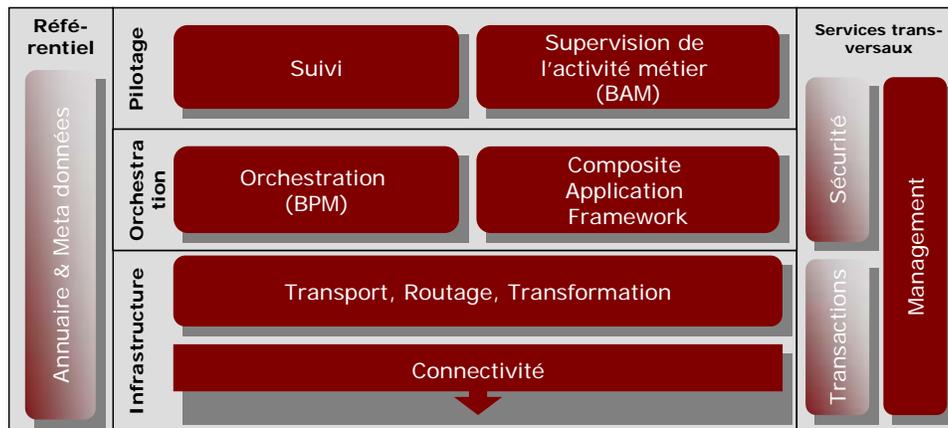


Figure 58 - Le modèle SOA de webMethods

Initialement présent sur le marché de l'EAI/B2B, l'offre de webMethods n'éprouve aucune difficulté à assurer une présence significative sur les couches basses du modèle, de même que sur l'orchestration, grâce au produit Modeler, et sur le suivi des flux et des processus. Ses rachats le positionnent efficacement sur le segment du BAM et du Composite Application Framework. Les services transversaux font l'objet d'un bon niveau d'implémentation.

6. Conclusion

Faut-il se réjouir de l'arrivée des SOA ou s'en méfier ? Faut-il s'enthousiasmer ou afficher un certain scepticisme ? De nombreux analystes n'ont pas manqué de relever que ces concepts n'étaient finalement qu'une façon de faire du neuf avec du vieux et, globalement, d'empiler une nouvel ensemble technologique au-dessus de strates à peine fossilisées.

Néanmoins, l'engouement que font naître ces concepts et ces technologies ne saurait être qu'un artefact médiatique destiné à remplir des pages de journaux (sinon une initiative telle que ebXML, qui présente un package technologique similaire, promu par les mêmes acteurs, et a fait l'objet d'une communication comparable, se serait davantage imposée). Pourquoi et comment le modèle technologique des Services Web est-il en passe de s'imposer là « facilement » où d'autres initiatives semblables, comme ebXML, ont connu et connaissent encore certaines difficultés ?

Le public des DSI, des architectes et urbanistes, des analystes et développeurs, a largement mûri depuis quelques années : on sait que toute nouveauté n'est pas bonne à prendre. Cette forte conjonction entre l'offre et la demande résulte donc d'espoirs et d'attentes forts, dans la capacité à organiser et à interopérer, dans cette combinaison de progrès technologique réel et de réduction des coûts, axe d'action idéal. Encore plus intéressantes sont les promesses d'une gouvernance accrue du système d'information par un apport simultané de cohérence et de flexibilité.

On sait également que les standards et technologies ne sont pas encore complètement finalisés, et que la réussite de l'entreprise sera, d'une certaine façon, tributaire de l'organisation que l'on parviendra à construire, en fonction également du périmètre départemental ou globalisant de la SOA dans l'entreprise (cette capacité à s'organiser autour de la technologie n'est pas l'apanage des SOA : on retrouve cette question dès lors qu'on cherche à construire des systèmes d'entreprise), en gardant à l'esprit que les éléments de ROI n'apparaîtront qu'avec la répétition des projets et la réutilisation qui en résulte. La capacité à prendre immédiatement du recul pour organiser avant de mettre en œuvre devient dès lors déterminante, se généralise à d'autres domaines émergents de l'industrie informatique, tels que le Master Data Management et devient ainsi l'un des principaux sujets d'investigation.

Glossaire

AS1 (Applicability Statement 1) : spécification pour les échanges EDI via e-Mail (SMTP), éditée par le groupe de travail EDIINT (EDI over the Internet) de IETF (Internet Engineering Task Force).

AS2 (Applicability Statement 2) : spécification pour les échanges EDI via HTTP et HTTPS, éditée par EDIINT.

Assertion SAML : méthode générique de description des informations de sécurité relatives à l'utilisateur ou à l'application.

J2EE (Java 2 Enterprise Edition) : ensemble d'interfaces de programmation orientées objet proposant un modèle pour l'implémentation des applications d'entreprise.

JAAS (Java Authentication and Authorization Services) : interface de programmation J2EE destinée à renforcer le contrôle d'accès aux ressources (applications, services...).

LDAP (Lightweight Directory Access Protocol) : technologie d'accès hiérarchique à toute forme de ressource hébergée dans un annuaire d'entreprise. Notamment utilisé en matière de sécurité pour stocker les listes d'utilisateurs et de groupes d'utilisateur avec leurs droits d'accès, ainsi que pour le SSO (*Single Sign-On*).

SAML (Security Assertion Markup Language) : protocole véhiculant des *assertions* dédiées à l'authentification des utilisateurs. Encapsulé dans les messages SOAP.

Sarbanes Oxley Act : Loi passée aux Etats-Unis en 2002 pour restaurer la confiance des investisseurs dans la santé des entreprises. Impose notamment des standards d'audit et de reporting, qui imposent de tracer les transferts de données. L'organisation centralisée de ces échanges et la conservation d'une trace devient ainsi un enjeu majeur pour les entreprises.

SMTP (Simple Mail Transfer Protocol) : protocole dstandardisé utilisé pour l'envoi et la réception d'e-Mails.

SOAP (Simple Object Access Protocol) : protocole XML sur la base duquel sont réalisés les échanges entre Services Web.

SSO (Single Sign-On) : technologie permettant la saisie unique et la correspondance entre informations d'accès utilisateur, pour éviter à celui-ci d'avoir à s'identifier à nouveau dès qu'il accède à de nouvelles ressources (applications ou services).

UDDI (Universal Description, Discovery and Integration) : technologie d'annuaire essentiellement utilisée dans le monde des services Web, pour stocker notamment leur description WSDL.

WS-Adressing : proposition conjointe de BEA, IBM, Microsoft, SAP et Sun pour véhiculer des messages SOAP de façon bidirectionnelle et indépendamment de l'infrastructure de transport, en environnement distribué (intégration multi-points)

WS-AtomicTransaction : voir WS-Coordination.

WS-Basic Profile : premier "profil" issu des travaux de la WS-I Organization. Normalisation de l'ensemble de spécifications concernant les protocoles de fondation du modèle des Web Services (WSDL 1.1, UDDI 2.0, SOAP 1.1, XML-Schemas 1.0).

WS-BusinessActivity : voir WS-Coordination.

WS-Coordination : proposition rédigée par BEA, IBM et Microsoft, spécifiant les modes de gestion des transactions pour les Services Web. Recouvre deux langages : WS-AtomicTransaction, dédié à la couverture des transactions atomiques ACID (domaine de confiance unique), et WS-BusinessActivity pour le support des transactions longues (multiples domaines de confiance).

WS-Eventing : standard qui vise à standardiser les mécanismes évènementiels de publication/abonnement (publish & subscribe) utilisés au niveau de la couche de transport et de routage de l'information.

WS-Federation : proposition de standard pour gérer les relations de confiance et le partage des identités entre partenaires.

WS-I Organization : association d'acteurs de l'industrie informatique, tels qu'IBM et Microsoft, vouée à créer des spécifications standard, normalisées et universellement utilisables pour les Web Services.

WS-Policy : proposition visant à décrire le niveau de sécurité attendu pour chacun des partenaires. Il s'agit ainsi d'une forme de gestion de contexte appliquée à la sécurité, fonctionnalité aujourd'hui couverte (de façon non standard) par de nombreux modules de gestion des partenaires B2B (Tibco Business Connect, webMethods Trading Networks...).

WS-Reliability : initiée par Sun, Oracle et Fujitsu, WS-Reliability est une spécification qui vise à assurer connexion inter-applicative à base de Web Services qui garantisse la bonne délivrance et la délivrance unique de l'information (ce qu'un middleware asynchrone est par ailleurs tenu d'assurer aujourd'hui de façon standard).

WS-ReliableMessaging : protocole construit conjointement par BEA, IBM, Microsoft et Tibco. Il spécifie les modes de délivrance garantie et unique de données entre applications distribuées.

WS-Security : extension de SOAP réalisée par IBM, Microsoft et Verisign pour sécuriser les échanges entre Web Services. Les informations sont véhiculées en clair. Couvre la signature, le chiffrement, les certificats, les jetons d'authentification.

WS-Trust : proposition de langage visant à standardiser les modèles de confiance entre partenaires.

WSDL (Web Services Description Language) : protocole XML de description des interfaces métier et des protocoles techniques de requête et de réponse à un Web Services.

XML Encryption : standard spécifiant le chiffrement d'un message XML.

XML Schemas : standard de définition des structures de données et méta-données.

XML Signature : standard spécifiant les modes d'ajout d'une signature numérique à un message XML.

XSLT (XML Stylesheet Transformation) : langage XML standardisé utilisé pour stocker la définition et pour exécuter les transformations de messages de format à format.

