Le contrôle de Qos pour les services web

Introduction :

La vision future de l'Internet est de transformer le réseau d'informations en un réseau de services. Cette évolution du monde informatique a entraîné le développement de nouveaux paradigmes d'interaction entre différentes applications. En effet, les services Web sont des technologies émergentes et prometteuses pour le développement, le déploiement

et l'intégration d'applications Internet. Ces technologies, basées sur XML, fournissent une infrastructure pour décrire WSDL: découvrir UDDI et invoquer SOAP des services. Un des avantages majeurs des services

Web par rapport middlewares traditionnels (CORBA, DCOM et XML-RPC) est l'apport de l'interopérabilité sur Internet. Les services Web ont pour vocation de favoriser une architecture orientée service, intégrant des systèmes complexes hétérogènes, fortement distribués et pouvant coopérer sans recourir à une intégration spécifique et coûteuse. Ce sont des applications accessibles sur Internet réalisant chacune une tâche spécifique. Toutefois, pour certains types d'applications, il est nécessaire de combiner un ensemble de services web élémentaires en des services Web composites (ou agrégés) dans le but de répondre à des exigences plus complexes et pour ne former, du point de vue de l'utilisateur, qu'un seul service Web.

Cependant, la notion de Qualité de Service (ou QoS) qui s'intéresse à la qualité de la relation entre un service et ses clients constitue un enjeu crucial. En fait, bien qu'il existe déjà d'importants travaux autour des compositions de services, qui ont notamment permis l'élaboration du standard BPEL4WS, le problème de la gestion de la QoS dans les compositions de services manque de solutions flexibles, réutilisables et offrant un degré d'abstraction approprié.

En effet, l'expressivité du langage BPEL vise uniquement les aspects fonctionnels des compositions. Or, dans le contexte des architectures orientées services où les acteurs se connaissent peu, les enjeux liés à la garantie de la qualité de service sont aussi fondamentaux que ceux liés à l'assemblage fonctionnel des services.

La notion qualité de service fait référence à diverses préoccupations telles que le temps de réponse ou encore la disponibilité. Le temps de réponse constitue est un facteur important dans le domaine des services web car sa prise en charge induit celle de la disponibilité au moment actuel. Effectivement, le fait d'invoquer un service web pouvant se trouvé n'importe où sur internet et obtenir une réponse, sous entend que ce dernier est disponible.

Ainsi, la complexité grandissante de la qualité de service, plus particulièrement du temps de réponse et son intérêt, tant pour les fournisseurs que pour les clients, nécessitent le développement d'outils permettant sa gestion, car la dégradation de cette dernière peut engendrer de sérieuses conséquences dont un impact économique important.

Le travail présenté dans cette thèse se situe justement dans le cadre de développement d'un Framework pour la vérification de contraintes qualité de service d'une composition de services web, plus précisément le temps de réponse.

Le présent mémoire, qui expose ce travail, est composé de quatre chapitres structurés comme suit :

- Dans le premier chapitre, on va présenter les services web et l'architecture SOA.
- Le deuxième chapitre sera consacré a l'étude du contrôle de qualité de service.

- Le troisième chapitre sera consacré au calcul des valeurs de Qos pour les SW composite.
- Dans le quatrième chapitre, on va parler sur les langages de composition et on va voir l'implémentation de l'application, en décrivant l'environnement logiciel et notre travail.

Chapitre 1 :Les services web et l'architecture SOA

1-Introduction :

Avec la grande utilisation du web, les chercheurs ont développé des logiciels pour assurer et simplifier la communication entre les machines et les applications connectées via le réseau, ces logiciels sont appelés «web services».

Dans ce chapitre, nous présentons les concepts de base de l'architecture orientée service (AOS).

2-Les services web :

Historique :

La légende raconte que c'est Bill Gates, alors président de Microsoft le premier qui a utilisé le terme Web Services. Il l'aurait fait le 12 Juillet 2000 au cours de Microsoft Professional Developers Conference à Orlando. Même si cette légende est controversée, il est plus certains en revanche que c'est chez Microsoft que ces mots ont été utilisés la première fois en les associant à SOAP, XML, WSDL et UDDI.

Evidement, tout ne s'est pas fait en jour, et la naissance des Services Web et des technologies qui les accompagnent remontent à un peu plus loin que cette date. En réalité, l'histoire commence en 1975 lorsque l'informatique souffrait encore de peu de standardisation et que les constructeurs et éditeurs se rendent compte de la nécessité d'uniformiser les échanges de données. Ils firent alors les vœux pieux de "l'interopérabilité" afin de standardiser la communication entre application au travers d'un réseau. C'est la naissance l'EDI (Electronic Data Interchange ou Echange de Données Informatisées), l'ancêtre des Web Services.

2-1définition :

On peut trouver plusieurs définitions des services Web tel que :

Citation1 : W3C

Un service Web est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en XML. Sa définition peut être découverte par d'autres systèmes logiciels. Les services Web peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles Internet.

Citation2 : Dico du Net

Une technologie permettant à des applications de dialoguer à distance via Internet indépendamment des plates-formes et des langages sur lesquels elles reposent.

Citation3 : Wikipédia

Un service Web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et en temps réel.

En d'autres termes, un service Web est tout simplement un programme accessible au moyen d'Internet, qui utilise un système de messagerie standard XML, et **n'est lié à aucun système d'exploitation ou langage de programmation !**

2-2 L'architecture d'un service web :

Les services Web reprennent la plupart des idées et des principes du Web (HTTP, XML), et les appliquent à des interactions entre machines. Comme pour le **World Wide Web**, les services Web communiquent via un ensemble de technologies fondamentales qui partagent une architecture commune. Ils ont été conçus pour être réalisés sur de nombreux systèmes développés et déployés de façon indépendante. Les technologies utilisées par les services Web sont HTTP, WSDL, REST, XML-RPC, SOAP et UDDI.

REST

REST :est une architecture de services Web. Élaborée en l'an 2000 par *Roy Fiedling*, l'un des créateurs du protocole HTTP, du serveur Apache HTTPd et d'autres travaux fondamentaux, REST est une manière de construire une application pour les systèmes distribués comme le World Wide Web.

XML-RPC

XML-RPC est un protocole simple utilisant XML pour effectuer des messages RPC. Les requêtes sont écrites en XML et envoyées via HTTP POST. Les requêtes sont intégrées dans le corps de la réponse HTTP. XML-RPC est indépendant de la plate-forme, ce qui lui permet de communiquer avec diverses applications. Par exemple, un client Java peut parler de XML-RPC à un PerlServer !

SOAP

SOAP (Simple object Access Protocol) est un protocole standard de communication. C'est l'épine dorsale du système d'interopérabilité. SOAP est un protocole décrit en XML et standardisé par le W3C. Il se présente comme une enveloppe pouvant être signée et pouvant contenir des données ou des pièces jointes.
Il circule sur le protocole HTTP et permet d'effectuer des appels de méthodes à distance.

WSDL

WSDL est un langage de description standard. C'est l'interface présentée aux utilisateurs. Il indique comment utiliser le service Web et comment interagir avec lui. WSDL est basé sur XML et permet de décrire de façon précise les détails concernant le service Web tels que les protocoles, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie et les exceptions pouvant être envoyées.

UDDI

UDDI (Universal Description, Discovery and Integration) est un annuaire de services. Il fournit l'infrastructure de base pour la publication et la découverte des services Web. UDDI permet aux fournisseurs de présenter leurs services Web aux clients.

Les informations qu'il contient peuvent être séparées en trois types :

- les pages blanches qui incluent l'adresse, le contact et les identifiants relatifs au service Web
- les pages jaunes qui identifient les secteurs d'affaires relatifs au service Web ;
- les pages vertes qui donnent les informations techniques.

Nous allons étudier plus en détail, ces trois dernières technologies.

3-L'architecture orientée services:

L'architecture orientée services est une architecture logicielle visant à mettre en place un système d'information constitué de services applicatifs indépendants et interconnectés.

L'architecture SOA permet la réutilisation des applications et des services existants, ainsi de nouveaux services peuvent être créés à partir d'une infrastructure informatique des systèmes déjà existante, en leur offrant une interopérabilité entre applications et technologies hétérogènes. L'objectif d'une architecture SOA est de décomposer une fonctionnalité en un ensemble de services et de décrire leurs interactions.

Dans une architecture SOA, les fournisseurs de services publient (ou enregistrent) leurs services dans un annuaire de services (qui peut être publique ou local à un réseau d'entreprise par exemple). Cet annuaire est utilisé par les utilisateurs (i.e. les clients de services) pour trouver des services vérifiant certains critères ou correspondant à une certaine description. Si l'annuaire contient de tels services, il envoie au client les descriptions de ces services avec un contrat d'utilisation. Le client fait alors son choix, s'adresse au fournisseur et invoque le service Web.

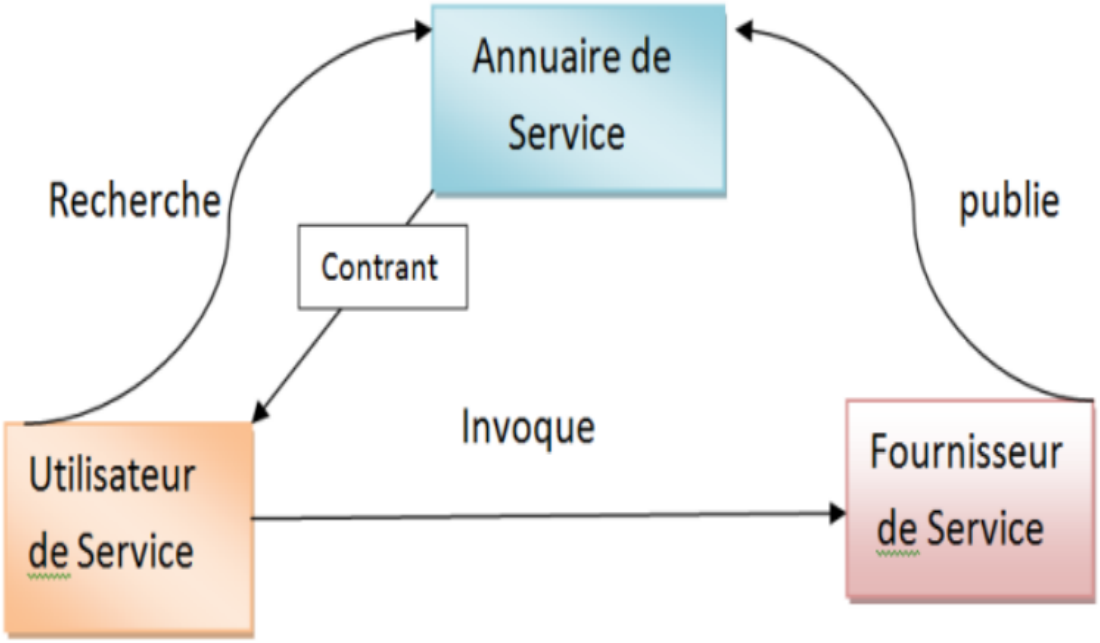


Figure 1.1

Cette architecture est caractérisée par :

1. un couplage faible entre les services : implique qu'un service n'appelle pas directement un autre service .En effet les interactions sont gérées par une fonction d'orchestration.

2. la réutilisation d'un service et plus facile, du fait qu'il n'est pas directement lié aux autres services de l'architecture dans laquelle il évolue.

3. L'indépendance par rapport aux aspects technologiques : est obtenue grâce aux contrats d'utilisation associés à chaque service. En effet, ces contrats sont indépendants de la plate-forme technique utilisée par le fournisseur du service.

4. la découverte des services disponibles, afin d'être sûr que les utilisateurs potentiels découvriront le service dont ils ont besoin.

5. La mise à l'échelle : est rendue possible grâce à la découverte et à l'invocation de nouveaux services lors de l'exécution.

4- Fonctionnement des services Web

Le fonctionnement des services Web s'articule autour de trois acteurs principaux illustrés par le schéma suivant :

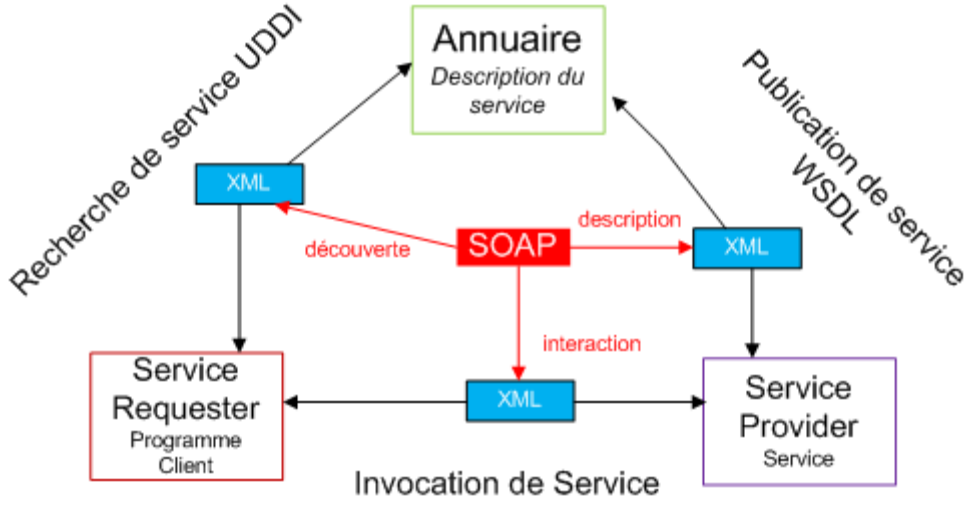


Figure 1.2

Décortiquons ce schéma :

A- Service provider service

Le fournisseur de service met en application le service Web et le rend disponible sur Internet.

B- Service requester programme client

C'est n'importe quel consommateur du service Web. Le demandeur utilise un service Web existant en ouvrant une connexion réseau et en envoyant une demande en XML (REST, XML-RPC, SOAP).

C- Annuaire service registry

Le registre de service est un annuaire de services. Le registre fournit un endroit central où les programmeurs peuvent publier de nouveaux services ou en trouver. Les interactions entre ces trois acteurs suivent plusieurs étapes :

- **La publication du service** : le fournisseur diffuse les descriptions de ses services Web dans l'annuaire.
- **La recherche du service** : le client cherche un service particulier, il s'adresse à un annuaire qui va lui fournir les descriptions et les URL des services demandés afin de lui permettre de les invoquer.
- **L'invocation du service** : une fois que le client récupère l'URL et la description du service, il les utilise pour l'invoquer auprès du fournisseur de services.

4.1-Description en couche des services Web

Les services Web emploient un ensemble de technologies qui ont été conçues afin de respecter une structure en couches sans être dépendante de façon excessive de la pile des protocoles. Cette structure est formée de quatre couches majeures :

Découverte de services	UDDI
Description de services	WSDL
Communication	SOAP

Transport	HTTP

Couches technologiques des services Web.

- Le transport de messages XML-RPC ou SOAP est assuré par le standard HTTP.
- SOAP ou XML-RPC prévoit la couche de communication basée sur XML pour accéder à des services Web.
- La description d'un service Web se fait en utilisant le langage WSDL. WSDL expose l'interface du service.
- La publication et la découverte des services Web sont assurées par le biais du référentiel UDDI. Un référentiel UDDI est un catalogue de services Web.

A-Couche transport

Cette couche est responsable du transport des messages XML échangés entre les applications. Actuellement, cette couche inclut HTTP, SMTP, FTP, et de nouveaux protocoles tels que BEEP.

B-Couche communication

Cette couche est responsable du formatage des données échangées de sorte que les messages peuvent être compris à chaque extrémité. Actuellement, deux styles architecturaux totalement différents sont utilisés pour ces échanges de données. Nous avons d'un côté l'architecture orientée opérations distribuées (protocoles RPC) basée sur XML et qui comprend XML-RPC et SOAP et de l'autre côté une architecture orientée ressources Web, REST (Representational

State Transfer) qui se base uniquement sur le bon usage des principes du Web (en particulier, le protocole HTTP).

C-Couche description de service

Cette couche est responsable de la description de l'interface publique du service Web. Le langage utilisé pour décrire un service Web est WSDL qui est la notation standard basée sur XML pour construire la description de l'interface d'un service. Cette spécification définit une grammaire XML pour décrire les services Web comme des ensembles de points finaux de communication (ports) à travers lesquels on effectue l'échange de messages.

D-Couche publication de service

Cette couche est chargée de centraliser les services dans un registre commun, et de simplifier les fonctionnalités de recherche et de publication des services Web. Actuellement, la découverte des services est assurée par un annuaire UDDI (Universal Description, Discrovery, and Integration).

5-Le cycle de vie d'une application à base d'architecture orientée services :

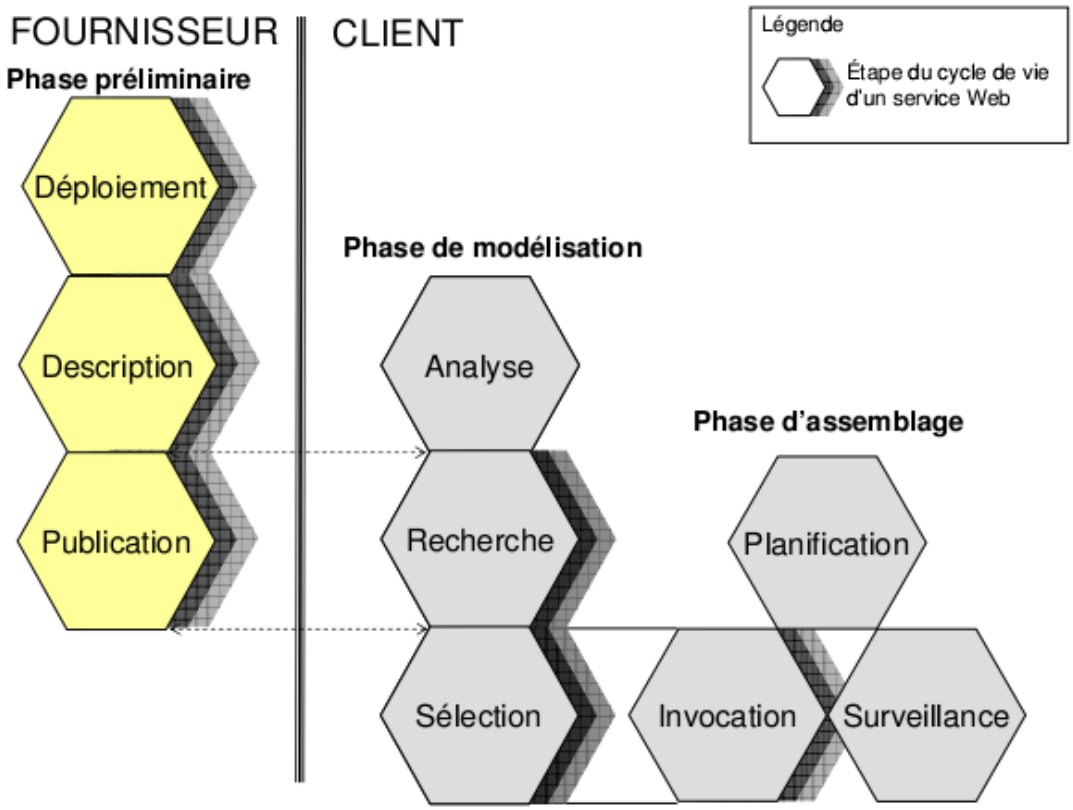


Figure 1.3

5-Conclusion :

L'architecture orientée services est une réponse très efficace aux problématiques que rencontrent les entreprises en termes de réutilisabilité, d'interopérabilité et de réduction de couplage entre les différents systèmes d'information. SOA est un style architectural pour créer des solutions logicielles basées sur les services. Le défi réalisé par SOA n'est pas seulement de créer les services, la vraie valeur de SOA vient lorsque des services réutilisables sont combinés pour former des processus métiers agiles et flexibles. Malheureusement cela ne va pas de soi. Il faut que les différents services soit basés sur des normes standardisés et interopérables syntaxiquement et

sémantiquement (en termes de propriétés fonctionnelles et non fonctionnelles). SOA est une architecture abstraite pour la conception des systèmes d'information, elle devra être implémentée par une technologie de service. La technologie de services la plus utilisée pour l'implémentation de l'architecture SOA est la technologie des services web.

Dans ce chapitre nous avons vu les bases de l'architecture SOA, ainsi que des services web. L'étude de Qos et du contrôle seront décrites dans le chapitre suivant.

Chapitre 2 : SOA étendu: composition et contrôle de QOS

1-Introduction:

L'une des questions de recherche des plus importantes dans le domaine des services Web est relative à la prise en compte des différents critères de la qualité (QoS, Quality of Service) de service dans la découverte et la sélection du meilleur service Web ayant une fonctionnalité donnée, afin de satisfaire au mieux le demandeur du service. En effet, on retrouve de plus en plus des services Web ayant une même fonctionnalité. Pour cela, plusieurs études proposent l'utilisation des différents critères non fonctionnels de qualité de services afin de les comparer et ainsi de choisir le meilleur parmi ceux disponibles. La QoS d'un service Web est définie comme étant une combinaison de plusieurs critères qui peuvent être quantitatifs tels que le temps de réponse, la disponibilité, ... ou qualitatifs tels que la sécurité, la disponibilité, etc.

2-Type de compositions des services web :

On trouver deux types de composition des services web :

2.1- Statique :

Ce type de composition peut être appliqué dans des environnements « stables » où les services Web participants sont toujours disponibles et où le comportement du service composite est le même pour tous les clients. La composition des services web prend place durant la période de conception. Les composants sont choisis, relient entre eux et, compilés et déployés. Le service composite ainsi obtenu fonctionnera bien tant que son environnement et les services qui le composent ne changent pas ou ne changent que rarement.

2.2- Dynamique :

Dans ce type de composition, les services Web à composer sont déterminés lors de l'exécution de la requête d'un client. Ils peuvent être déterminés selon les contraintes de chaque client, la disponibilité des services Web, etc. La composition dynamique apparaît la plus intéressante d'une part, elle promet d'être capable de faire face à un environnement très dynamique dans lequel des services apparaissent et disparaissent rapidement. D'autre part, elle permet de mieux satisfaire les besoins de chaque client en minimisant son intervention.

2.2.1-Manuelle :

La composition manuelle des services Web suppose que l'utilisateur génère la composition à la main via un éditeur de texte et sans l'aide d'outils dédiés.

2.2.2-Semi-automatique :

Les techniques de composition semi-automatiques sont un pas en avant en comparaison avec la composition manuelle, dans le sens qu'ils font des suggestions sémantiques pour aider à la sélection des services Web dans le processus de composition.

2.2.3-Automatique :

La composition totalement automatisée prend en charge tout le processus de composition et le réalise automatiquement, sans qu'aucune intervention de l'utilisateur ne soit requise.

La composition manuelle est, parmi les trois, la plus adaptable aux besoins de l'utilisateur, car il va tout définir à son goût depuis le début. Mais, malheureusement, elle nécessite des connaissances de bas niveau de

programmation. Étant donné que l'environnement dans lequel nous allons travailler est destiné aux utilisateurs non informaticiens cette catégorie de composition doit être rejetée.

Cependant, il faut maintenir le plus grand niveau de flexibilité possible afin que l'utilisateur puisse définir son propre processus. C'est pour cette raison que la composition automatique a aussi été exclue car l'utilisateur n'intervienne pas dans la définition du processus de composition.

Il nous reste la composition semi-automatique des services web. Cette composition étant à mi-chemin entre les deux antérieures est la plus adéquate dans la composition des services web.

L'utilisateur maintiendra certain contrôle sur le processus envisagé mais il n'aura pas besoin de connaissances de programmation et il pourra définir son propre processus en utilisant des outils graphiques pour modéliser et concevoir des services Web composites.

3 -Approches de composition des services web :

La composition peut être décrite sous deux angles :

3.1-Orchestration :

Une orchestration assemble les services web dans un processus métier exécutable qui doit être exécuté par un moteur d'orchestration.

L'orchestration de services Web (Figure 2.1) consiste à la programmation d'un moteur qui appelle un ensemble des services web selon un processus prédéfini. Ce moteur définit le processus dans son ensemble et appelle les services web (tant internes qu'externes à l'organisation) selon l'ordre des tâches d'exécution

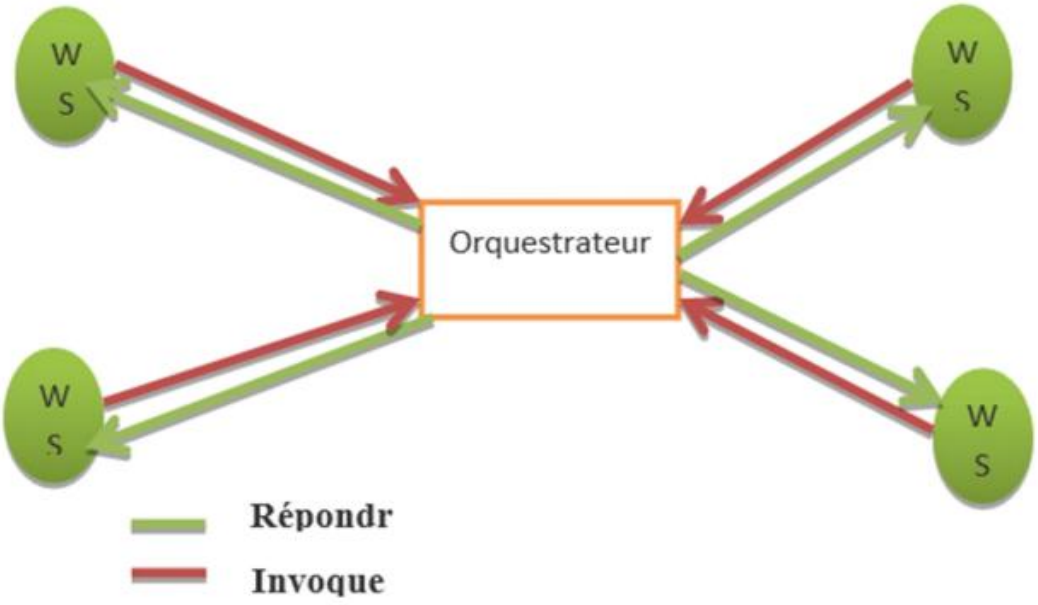


Figure 2.1

3.2-Chorégraphie :

La chorégraphie n'implique pas un contrôle centralisé, le contrôle est partagé entre les participants en interaction. Une orchestration représente un processus exécutable à exécuter par un moteur d'orchestration en un seul endroit, alors que la chorégraphie en essence représente une description de la façon de distribuer le contrôle entre les participants collaborent, à l'aide des plusieurs moteurs pour faire le travail.

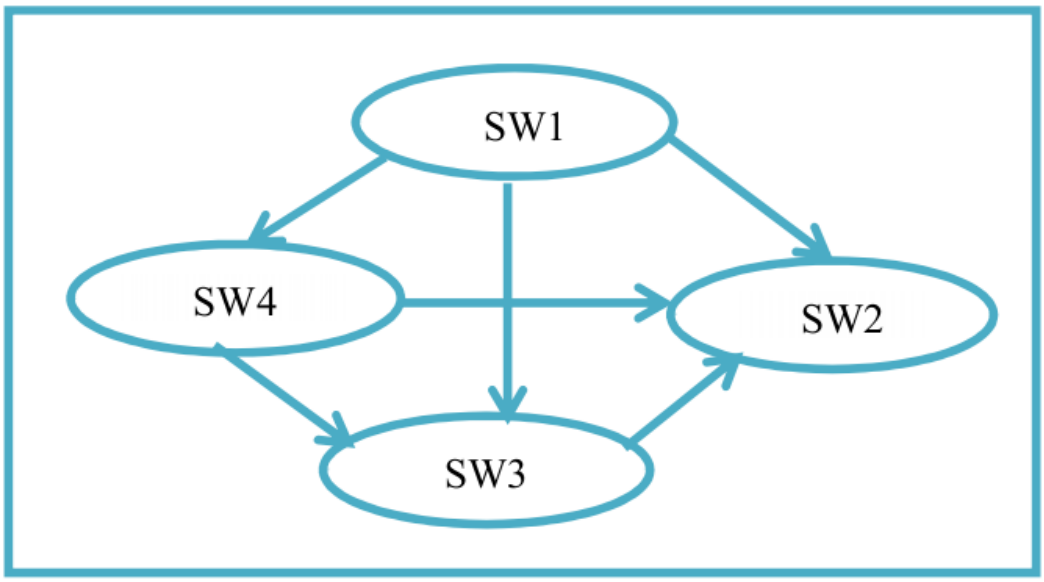


Figure 2.2

4- Le QoS (quality of service):

Les services web sont des systèmes logiciels qui répondent généralement à des besoins fonctionnels et des travers des besoins non fonctionnels Les besoins non fonctionnels des services web s'expriment généralement à travers la qualité de service

QoS est un ensemble de propriétés et caractéristiques d'une entité ou d'un service qui lui confèrent l'aptitude à satisfaire des besoins déclarés ou implicites. Ces besoins peuvent être liés à des paramètres tels que l'accessibilité, la disponibilité, le temps de réponse, le coût, la fiabilité, etc.

QoS se réfère à la capacité du service Web pour répondre à attendre invocations et de les exécuter au niveau correspondant à la mutuelle attentes à la fois son fournisseur et ses clients. Plusieurs facteurs de qualité qui reflètent les attentes des clients, tels que la disponibilité constante de service, la connectivité et haute la réactivité, la clé devient pour garder une entreprise compétitive et viable, car ils peuvent avoir un impact sérieux sur la prestation de services. QoS devient ainsi un important critère qui détermine la facilité d'utilisation du service et

l'utilité, les deux qui influencent la popularité d'un service particulier de Web, et une vente important et la différenciation point de rencontre entre les fournisseurs de services Web.

Delivering QoS sur Internet est essentiel et important en raison de sa nature dynamique et imprévisible. Applications avec des caractéristiques très différentes et les exigences en concurrence pour toutes sortes de ressources de réseau. Les changements dans la circulation motifs, la sécurisation des transactions commerciales critiques, et les effets de la défaillance de l'infrastructure, la faible performance des protocoles Web, et les problèmes de fiabilité plus le Web crée un besoin de normes Internet QoS. Souvent, les problèmes de qualité de service non résolus provoquent des applications transactionnelles critiques à souffrir de niveaux inacceptables de dégradation de la performance.

Traditionnellement, la qualité de service est mesurée par le degré auquel les applications, les systèmes, réseaux, et tous les autres éléments de la disponibilité de soutien de l'infrastructure informatique de services à un niveau de performance requis dans toutes les conditions d'accès et de charge.

Bien que des mesures de QoS traditionnelles puissent être appliquées, les caractéristiques d’environnements des services Web apporter à la fois une plus grande disponibilité des applications et la complexité accrue en termes d'accès et de gestion des services et donc imposer spécifique et intense demandes sur les organisations qui QoS doit répondre. Dans le contexte des services Web, QoS peut être considérée comme une assurance sur un ensemble de caractéristiques quantitatives.

Ceux-ci peuvent être définis sur la base du service fonctionnel et non fonctionnel important les propriétés de qualité qui incluent les questions de mise en œuvre et de déploiement ainsi que d'autres caractéristiques des services

importants tels que les compteurs de service et de coût, de performance métriques (par exemple le temps de réponse), les exigences de sécurité, l'intégrité, la fiabilité, l'évolutivité, et la disponibilité. Ces caractéristiques sont les exigences nécessaires pour comprendre le comportement global d'un service afin que d'autres applications et le service peut se lier à lui et l'exécuter dans le cadre d'un processus d'affaires.

Les éléments clés de la qualité de service de support dans un environnement de services Web sont résumés dans ce qui suit :

1. Disponibilité: La disponibilité est l'absence d'interruptions de service. Disponibilité représente la probabilité qu'un service est disponible. Des valeurs plus élevées signifient que le service est toujours prêt à l'emploi tandis que les valeurs plus petites indiquent

Imprévisibilité quant à savoir si le service sera disponible à un particulier temps. Sont également associés à la disponibilité est le temps à la réparation (TTR). TTR représente le temps nécessaire à la réparation d'un service qui a échoué. Idéalement plus petite valeurs de TTR sont souhaitables.

2. Accessibilité: L'accessibilité représente le degré avec lequel un service Web demande est servi. Elle peut être exprimée en tant que mesure de probabilité indiquant le taux ou le hasard d'une instanciation de service avec succès à un point dans le temps succès.

Un degré élevé d'accessibilité signifie qu'un service est disponible pour un grand nombre de clients et que les clients peuvent utiliser le service relativement facilement.

3. Conformité aux normes. Décrit la conformité d'un service Web avec

Normes. Le strict respect de corriger les versions de normes par le fournisseur d’un service est nécessaire pour le bon appel de services Web par le service

demandeurs. En outre, les fournisseurs de services doivent respecter les normes énoncées dans les accords de niveau de service entre demandeurs et fournisseurs de services.

4. Intégrité. Décrit le degré avec lequel un service Web exécute ses tâches selon sa description WSDL, ainsi que la conformité avec Service-Level Agreement (SLA). Un degré plus élevé d'intégrité signifie que la fonctionnalité d'un service est plus proche de sa description WSDL ou SLA.

5. Performance. La performance est mesurée en termes de deux facteurs: le débit et la latence. Débit représente le nombre de demandes de service Web à un compte tenu de la période de temps. Latence représente le temps écoulé entre l'envoi d'une demande et recevoir la réponse. Un débit plus élevé et une latence plus faible les valeurs représentent une bonne performance d'un service Web. Lorsque l'on mesure les volumes transaction / de demande traitées par un service Web, il est important de examiner si ceux-ci viennent dans un flux constant ou éclatent autour de particulier des événements comme l'ouverture ou la fermeture de la journée d'affaires ou de rushes saisonniers.

6. Fiabilité. La fiabilité représente la capacité d'un service de fonctionner correctement, et de manière cohérente et de fournir la même qualité de service en dépit du système ou les défaillances du réseau. La fiabilité d'un service Web est généralement exprimée en termes de nombre d'échecs de transactions par mois ou par année.

7. Extensibilité. Évolutivité se réfère à la capacité de servir systématiquement les demandes en dépit des variations du volume des demandes. Haute accessibilité du Web les services peuvent être obtenus en construisant des systèmes hautement évolutifs.

8. Sécurité. Sécurité concerne des aspects tels que l'authentification, l'autorisation, l'intégrité des messages, et la confidentialité. La sécurité a une

importance supplémentaire parce que le service Web invocation se produit sur Internet. La quantité de sécurité d'un service Web particulier requiert est décrite dans son accompagnement SLA, et les fournisseurs de services doivent maintenir ce niveau de la sécurité.

9. transactionnalité. Il y a plusieurs cas où les services Web nécessitent un comportement transactionnel et la propagation de contexte. Le fait qu'un Web particulier le service exige un comportement transactionnel est décrit dans son SLA qui l'accompagne, et les fournisseurs de services doivent conserver cette propriété.

Les critères de QdS représentent l'aspect non fonctionnel d'un service web et c'est à l'aide de ces critères que les exigences non fonctionnelles du demandeur de services peuvent être exprimées. Ces différents critères peuvent être classifiés selon leur déterminisme. Ainsi un critère est déterministe lorsque sa valeur est connue ou certaine quand le service web est invoqué (exemple : prix d'exécution, taux de pénalité) et il est non déterministe lorsque sa valeur est incertaine lorsque le service web est invoqué (exemple : durée d'exécution).

D'après une étude de la majorité des critères de QdS relatifs au temps, on peut juger que les critères de QdS liés à la performance tels que le temps de réponse, la latence et le débit

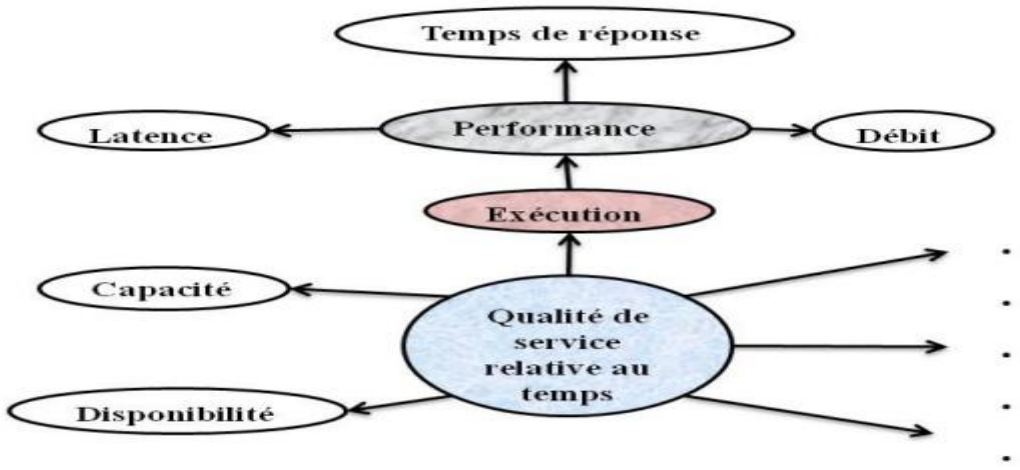


Figure 2.4

5- Le contrôle :

Le contrôle consiste en une vérification à l'exécution que les exigences, spécifié par les clients et par les fournisseurs de services, sont respectées lors de l'exécution. Les exigences peuvent évidemment être de nature très diverse. Il ya trois dimensions complémentaires au problème du contrôle coexistantes:

• contrôle basée sur les assertions

• contrôle basée sur les événements

• contrôle basée sur un historique de contrôle par Assertion-consiste à demander le processus BPEL d'utiliser un service de moniteur externe pour vérifier l'exactitude de certaines affirmations des points donnés de l'exécution du processus. Les affirmations qui doivent être contrôlées sont les prés et post conditions dérivée de la combinaison de ceux fournis par le fournisseur de services et ceux fournis par le client du processus. A partir d'une BPEL complète le processus - assertions Process- incontrôlées sont ajoutés au processus sous la forme des commentaires annotations. L'emplacement dans lequel ils sont ajoutés indique où, dans un traitement d'une assertion doit être vérifiée. Une fois une version annotée du non contrôlé procédé est disponible, elle est passée à travers

un algorithme de transformation appelée BPEL2BPEL, qui produit ce que nous appelons le processus contrôlé. L'algorithme ajoute le code BPEL au processus pour appuyer l'utilisation d'un service de moniteur externe qui vérifier la validité des assertions. Le code ajouté prépare le message à envoyer au moniteur externe, l'envoie, et vérifie la réponse du moniteur. Le message qui est envoyé au service de moniteur externe contient deux éléments: l'assertion à vérifier et les données sur lesquelles les assertions doivent être vérifiées .Ces données sont généralement constituées d'informations sur l'état interne du processus en cours d'exécution. En fonction de la mise en œuvre du moniteur externe, les données sur lesquelles vérifier les assertions pourraient également être obtenues ailleurs.

Cette approche a évidemment un impact important sur la performance, depuis l'exécution du processus est momentanément arrêté pour exécuter le contrôle. Cependant, il est tout simplement une technique qui est offert au concepteur du processus, qui peut être utilisé comme nécessaire. Par exemple, il peut être utilisé que de manière sporadique, ou même être éteint totalement ou partiellement à tout moment.

Sa force est que, puisque le contrôle est inséré dans la forme d'annotations, la logique métier reste distincte du contrôle la logique jusqu'au moment où l'BPEL2BPEL effectue la transformation qui tisse les deux pour produire une nouvelle version du processus contrôlé. La version originale n'est pas touchée. Deuxièmement, le processus résultant reste code BPEL pure et exécutable sur un moteur BPEL standard. L'inter-tissage devrait être en mesure de prendre en compte les différents besoins du contrôle des prenantes différentes et / ou des moments différents du cycle de vie du processus. Il est noté que, dans cette approche, il est plus facile de suivre des contrats fonctionnels, en ce qui concerne les contrats non-fonctionnels.

Le contrôle basé sur les événements entre en jeu quand il est nécessaire de vérifier la qualité non fonctionnelle d'un service ou d'une composition. Il se

compose d’une moins intrusive l’approche du contrôle où, parallèlement à l'exécution d'un processus d'affaires, composante du contrôle peut écouter les événements lancés par le moteur BPEL. Depuis cela se fait en parallèle avec l'exécution, presque pas d'impact sur les performances est attendu. De toute évidence, cette approche est étroitement liée à la mise en œuvre particulière du moteur BPEL utilisé. Par exemple, en utilisant Active BPEL, un BPEL open source moteur, il est possible d'écouter une série d'événements que le moteur fournit. Dans notamment, ces événements sont liés aux activités BPEL standard (à savoir invoquer, recevoir, etc.) et de leurs changements d'état.

Chaque activité peut être, par exemple, dans un état prêt à exécuter l'état, dans un état d'exécution, dans un état terminé ou dans un état irréprochable.

Une autre approche possible serait de positionner un service d'observateur entre le moteur BPEL et le monde extérieur et de surveiller le contenu du SOAP les messages entrant et sortant du système. Les deux approches doivent être plus enquête afin de découvrir les forces et les faiblesses de chacun. Au moment, il est clair que les deux travaux à un niveau inférieur par rapport au processus BPEL et l'approche basée sur l'affirmation dans le contrôle. En conséquence, une fois que le comportement erroné est observé, il est impossible de simplement geler l'exécution du processus à mettre en œuvre une action de récupération, comme la reconfiguration dynamique du processus. L'intervention dans l'exécution peut être obtenue en sélectionnant des points dans le processus dans lequel d'introduire une affirmation. Cette affirmation pourrait représenter une point de synchronisation entre l'exécution des processus et la collecte des événements. En ajoutant une affirmation, nous serions mélangeons efficacement le contrôle basée sur les événements avec sur la base des assertions du contrôle. Par conséquent, si l'affirmation est fausse, au moins l'un des formules contrôlés LTL doivent être faux, ce qui signifie des mesures de rétablissement devraient

être prises. le contrôle basée sur l'historique est une extension de la surveillance basée sur les événements. Par la collecte des événements dans un référentiel d'événements de l'histoire, il est possible de raisonner sur la qualité de service exigences qui traitent de l'histoire des exécutions de processus. Un exemple d'une telle exigence pourrait être «quatre-vingts pour cent des fois un processus va dans l'exécution il doit remplir en une minute ". L'approche basée sur des événements et de la base de l'histoire, approche doivent encore être étudiés en profondeur, mais il semble déjà clair que les trois doivent coexister pour obtenir le contrôle fonctionnelle et / ou non fonctionnel complet.

5-1 Les méthodes de contrôle :

5-1-1 La méthode du contrôle de Niveau de Service:

Méthode de contrôle de niveau de service est la méthode la plus fondamentale du contrôle des paramètres de qualité. Cette approche est l'une des premières solutions pour obtenir les valeurs des paramètres de qualité de service d'hébergement Web. Dans cette approche, le code du moniteur est entré dans le code client et le fournisseur de service web .Par exemple, mettre une minuterie sur le code client d'enregistrer une heure de la demande et son temps de réception, ou par encapsulation dans les fonctions opérationnelles du fournisseur de services client / web sont inspirés par l'architecture orientée aspect. Le contrôle des paramètres de qualité du niveau de service est parmi les méthodes les plus courantes en matière de contrôle de la qualité des paramètres qui est possible par le codage du côté client ou Du côté serveur ou les deux d'entre eux. Pour expliquer plus en détail cette question, nous porterons une attention pour mesurer le paramètre de temps de réponse en utilisant la méthode de contrôle de niveau de service. Une solution simple pour mesurer les caractéristiques d'un service web est de développer la fonction de service proxy en plus de leurs codes respectifs.

Service Web Proxy contient des codes que les conditions de liaison sont spécifiées dans l'interface de service. De ce fait, la complexité du réseau de communication et leurs détails restent caché par le client.

5-1-2 La méthode contrôle du niveau de communication :

Dans cette méthode, le contrôle est réalisé en interprétant les messages échangés entre les clients et les fournisseurs service Web. Les messages sont échangés dans les services Web par SOAP, protocoles HTTP, TCP / IP. Dans ces procédés, en analysant des messages de couche de communication échangés, on a tenté de collecter des données sur les paramètres de qualité. De ce fait, les valeurs de paramètres de qualité d'un service Web peuvent être obtenus. le contrôle des paramètres de qualité par l'analyse de la HTTP + TCP / protocole de communication IP est effectué.

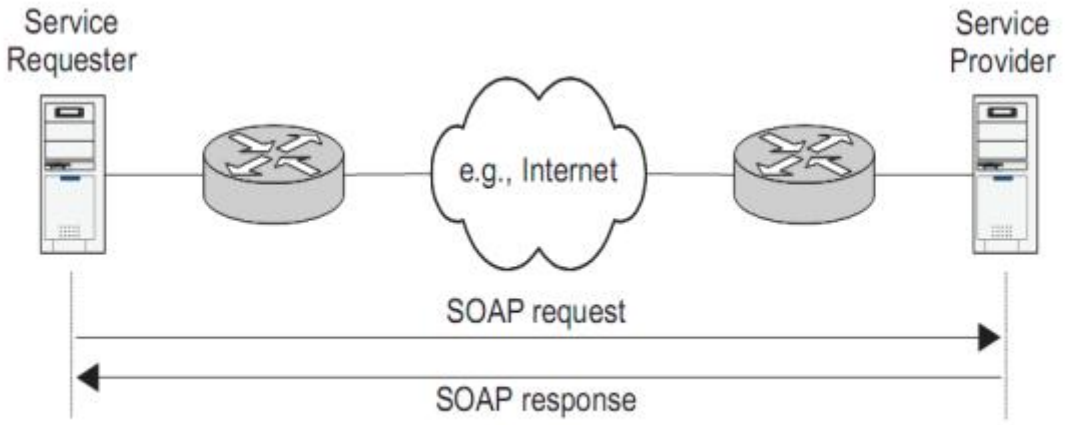


Figure 2.5

La figure 2.4 montre les couches de communication existent dans une simple relation entre le serveur et le client. Applicateur de service génère un message de demande SOAP et l'envoie au service fournisseur pour recevoir des services utilisant le protocole HTTP.

Ce message est être passé par plusieurs systèmes entre les deux parties.

Prestataire de services envoie le message de réponse SOAP par HTTP. Dans le mode d'envoi et de réception, tous les événements rencontrer envoi de données peut affecter les performances de services Web. Les problèmes incluent le surpeuplement, le manque de 'accès à l'hôte et le numéro de routage de ces problèmes peuvent être résolus par des protocoles de communication modernes.

Les protocoles de communication fournissent des informations en qui pourrait être utilisé pour contrôler les paramètres de qualité de bande de service.

5-1-3 méthode de contrôle de Niveau d'orchestration :

Dans cette méthode, le contrôle des paramètres de qualité est effectué sur les services BPEL utilisant Architecture fait orientée aspect.

Le contrôle des paramètres de qualité des supports le suivi de l'échantillon et de la classe. Utilisation de l'aspect orientée architecture permet la logique métier de service distinct de la logique de contrôle.

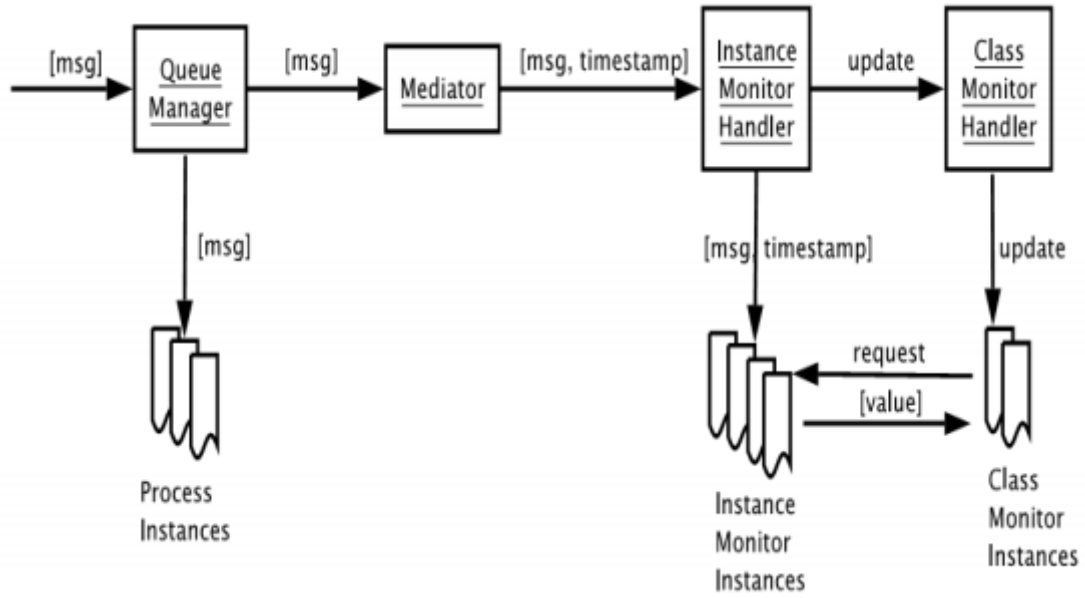


Figure 2.6

Par exemple, le contrôle de l'échantillon est effectué pour obtenir le comportement des bases de données des services Web et le contrôle de la classe est effectué pour obtenir le comportement du service de paiement en ligne des sites Internet.

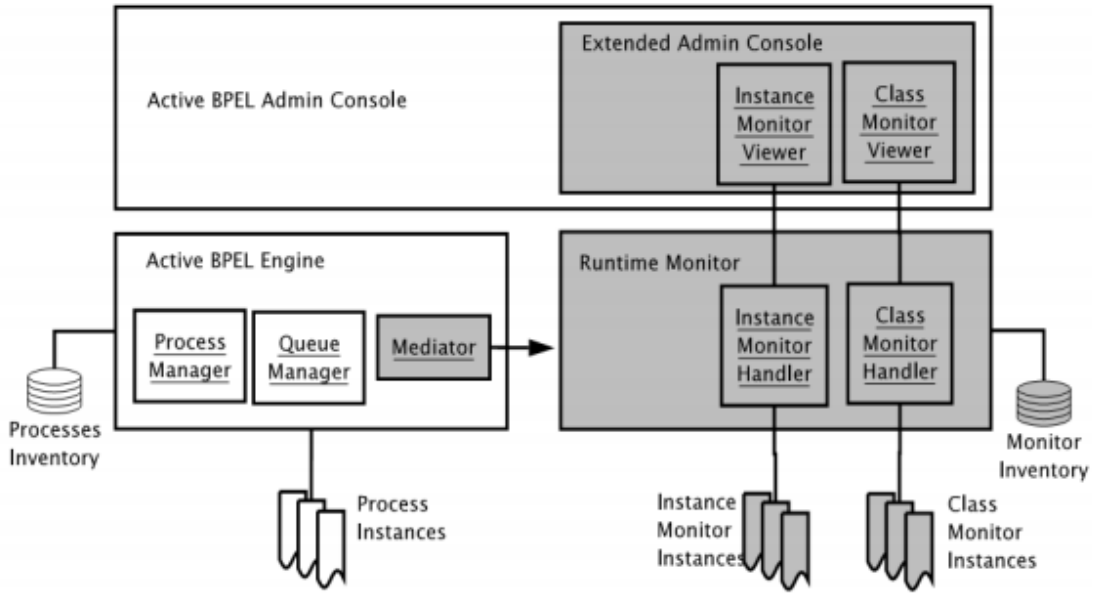


Figure 2.7

Ici, le contrôle des paramètres de qualité est effectuée par l'architecture orientée aspect et les valeurs de qualité des paramètres sont déterminés en analysant les messages entrée / sortie. Les données d'entrée / sortie sont envoyés et reçus par un processus.

Le contrôle peut être validé de trois façons par :

Le fournisseur de service lui-même (solution souvent par défaut).

Le client du service, qui doit alors mettre en œuvre un système de mesure adapté.

Un tiers de confiance qui gère le système de mesures et aide à l'interprétation.

5-2 L’objectif du contrôle :

Correction du système de l'exécution: il est capable d'analyser si les informations au moment de l'exécution correspondre aux valeurs attendues ou pas? En outre, cette information est nécessaire pour analyser les exigences des accords de niveau de service et de comparer les spécifications des systèmes.

En outre, les demandeurs de services, en utilisant les informations obtenu à partir du contrôle, vérifiera les valeurs avec leur valeurs attendues, qui sont les mêmes valeurs, inscrites dans la validation du niveau service. Le résultat déterminera exactement le niveau de satisfaction de la clientèle du service. Cette information est très utile pour le fournisseur de services et, car avec cette information, il peut s'adapter à la nouvelle condition et couvrent les faiblesses résultaient du contrôle.

5-3 La nécessité du contrôle :

Une fois que les services et les processus d'affaires deviennent opérationnels, leurs progrès doit être géré et contrôlé pour obtenir une vision claire de la façon dont les services effectuent au sein de leur environnement opérationnel, prendre des décisions de gestion, et d'effectuer le contrôle des actions pour modifier et ajuster le comportement des services Web- applications activées. Le contrôle de niveau de service est une méthodologie rigoureuse pour établir des niveaux acceptables de service qui répondent aux objectifs commerciaux, les processus et les coûts.

La phase du contrôle de service se préoccupe de la mesure de niveau de service; suivi la procédure en boucle fermée continue et de mesure, de contrôler, rapports, et l'amélioration de la qualité de service des systèmes et des applications fournis par le service de la solution orientée. Le contrôle du service

implique plusieurs activités distinctes, y compris exploitation forestière et de l'analyse des détails de l'exécution du service, l'obtention d'affaires métriques par l'analyse des données d'exécution de service, détection des situations d'affaires qui nécessitent l'attention de la direction, la détermination des mesures de contrôle appropriées pour prendre en réponse à la situations d'affaires, que ce soit pour atténuer un risque ou de prendre une occasion, et en utilisant les données de performance du service historique pour l'amélioration continue des services.

La phase du contrôle du service cible l'évaluation continue des niveaux de service objectifs, le contrôle de la couche de services pour la disponibilité et la performance, la gestion les politiques de sécurité, le suivi inter connectivité des composants faiblement couplés, l'analyse de la cause et corriger les problèmes. Pour atteindre cet objectif, le service contrôle l'exige qu'un ensemble de paramètres de qualité de service sont recueillies sur la base de SLA, étant donné qu'un SLA est une compréhension des attentes de service. En outre, les charges de travail doivent être contrôlées par le fournisseur de service pour assurer que la performance promise niveau est livré, et de prendre les mesures appropriées pour remédier à la non-conformité avec un SLA, comme des priorités et l'allocation des ressources.

Pour déterminer si un objectif a été atteint, la qualité de service SLA-disponible et les paramètres sont évalués en fonction des données mesurables sur un service (par exemple le temps de réponse, le débit, la disponibilité, etc.), la performance durant des périodes déterminées, et périodique d'évaluations. SLA comprennent d'autres objectifs observables, qui sont utiles pour le contrôle du service. Ceux-ci incluent la conformité avec les offres de niveau de service différenciée, à savoir fournir la qualité de service différenciés pour les différents types de clients, le niveau de service individualisé des offres et des demandes de police qui assure que les demandes d'immatriculation par les client reste dans une

limite prédéfinie. Tous ceux-ci doivent également être contrôlée et évaluer. Un aspect clé de la définition des objectifs mesurables est de fixer des seuils d'alerte et des alertes pour les échecs de conformité. Par exemple, si le temps d'une réponse particulière d'un service se dégrade alors le client peut être automatiquement acheminé vers une sauvegarde d'un service.

6- Services impliqués dans le suivi de la conformité :

Au cours du processus de passation des marchés, après les principaux éléments de la SLA sont d'accord sur, le client et le fournisseur peuvent définir des tiers à laquelle les tâches de contrôle SLA peuvent être déléguées. Lorsque la SLA est finalisé, à la fois fournisseur et le client font de la SLA document disponible pour le déploiement. Le service de déploiement est responsable de vérifier la validité de la SLA et de le distribuer en totalité ou en partie à la soutenir les parties. Figure 6.1 illustre les services impliqués dans le contrôle de la conformité lorsque plusieurs parties sont impliquées.

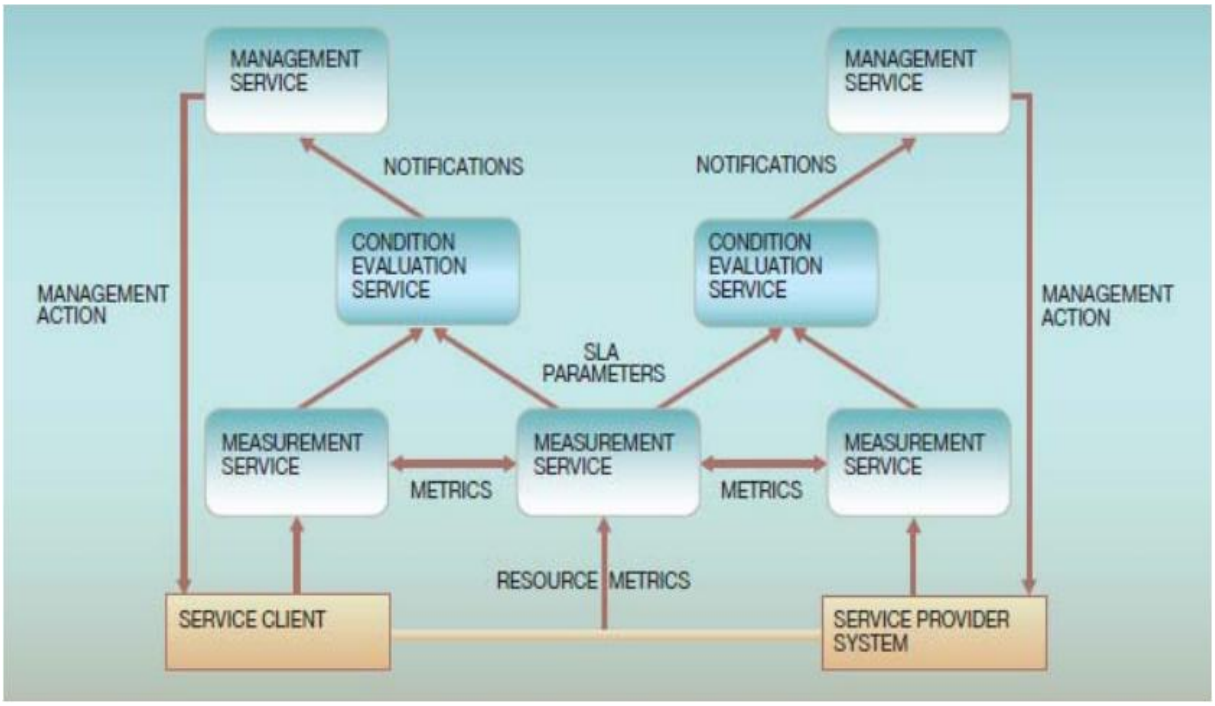


Figure 2.6

Les services qui peuvent être externalisées à des tiers sont des services soit mesure ou les services conditions d'évaluation.

Le service de mesure maintient des informations sur le système actuel la configuration et l'exécution des informations sur les mesures qui font partie de la SLA. Il mesures des paramètres SLA, tels que la disponibilité ou le temps de réponse, soit à partir de l'intérieur, en récupérant des métriques de ressources directement à partir de ressources gérées, ou à partir de l'extérieur de la domaine de fournisseur de services, par exemple, en sondant ou interception client transactions. Un service de mesure peut mesurer tout ou une partie de la SLA paramètres. Plusieurs services de mesure peuvent mesurer simultanément la même métrique, par exemple, un service de mesure peut être situé dans le domaine du fournisseur tandis qu'un autre service de mesure sonde le service offert par le fournisseur à travers Internet à partir de divers endroits.

Comme cela est représenté à la figure 6.1, les services de mesure peuvent être montés en cascade, soit un troisième service de mesure peut être utilisé pour les données globales calculées par d'autres services de mesure. De cette façon, nous nous référons à des mesures qui sont extraites directement à partir de ressources gérées comme des métriques de ressources. Les métriques composites, en revanche, sont créé par l'agrégation de plusieurs ressources (ou d'autres composites) métriques selon un algorithme spécifique, comme la moyenne une ou plusieurs mesures sur une quantité spécifique de temps ou en les décomposant en fonction de critères spécifiques (par exemple, 5 pour cent, minimum, maximum, moyenne, la médiane, etc.). Cela se fait habituellement par une mesure de service à l'intérieur du domaine d'un fournisseur de services, mais peut être confiée à un tiers de service mesure aussi bien.

La condition de service d'évaluation est responsable du contrôle du respect des paramètres SLA lors de l'exécution avec le convenu objectif de niveau de service (SLO) en comparant les paramètres mesurés par rapport aux seuils

définis dans le SLA et avertir les services de gestion du client et le fournisseur. Il obtient

Les valeurs mesurées des paramètres de SLA à partir d'un ou plusieurs services de mesure et les tests contre les garanties données dans le SLA. Cela peut être fait à chaque fois qu'une nouvel valeur est disponible, ou périodiquement.

Enfin, à la fois le client de service et le fournisseur ont un service de gestion.

À la réception d'une notification, le service de gestion prend des mesures appropriées pour corriger un problème, tel que spécifié dans la SLA. Le but principal de la direction service est d'exécuter des actions correctives au nom de l'environnement géré si une condition de service d'évaluation découvre qu'un terme d'un SLA a été violé.

7-Conclusion :

L'objectif de la composition des services web est de créer de nouvelles fonctionnalités en combinant des fonctionnalités offertes par d'autres services existants en utilisant plusieurs types et approches telles que la composition statique. La qualité de service ou bien les propriétés non fonctionnelles, les différentes approche du contrôle et ses objectifs on été aussi noté dans ce chapitre.

Chapitre 3:Contrôle de QOS pour une composition statique

1-Introduction :

Les entreprises peuvent utiliser un service simple pour accomplir une tâche spécifique de l'entreprise, comme la facturation ou le contrôle des stocks. Toutefois, pour les entreprises d'obtenir la pleine avantages des services Web, des processus d'affaires et des services Web transactionnels comme la fonctionnalité est nécessaire qui est bien au-delà que l'on trouve dans le service Web de prestations d'information. Lorsque les entreprises doivent composer plusieurs services ensemble pour créer un processus d'affaires tels que la commande personnalisée, soutien à la clientèle, l'approvisionnement et Le soutien logistique, ils ont besoin d'utiliser des services Web complexes. Les services complexes ou composites, impliquent généralement l'assemblage et l'invocation de nombreux services préexistants éventuellement trouvé dans diverses entreprises pour compléter une interaction d'affaires en plusieurs étapes.

Considérons par exemple une agence de voyage utilisant des services d'achat de billets d'avion et de réservation hôtelière. Le premier appartient à une compagnie aérienne et le deuxième à une chaine hôtelière. Une troisième entreprise d'agence de voyages veut mettre en place un service web d'organisation de formules de voyage selon une liste de critères ou de préférences (vol avec réservation d'hôtel). Au lieu de mettre en place un nouveau service, elle veut combiner et utiliser un mécanisme de gestion de préférences utilisant les services des deux compagnies précédentes. On parle d'agrégation de services.

La figure 3.1 montre les services utilisés par l'agence de voyage et les intervenants dans le processus de réservation :

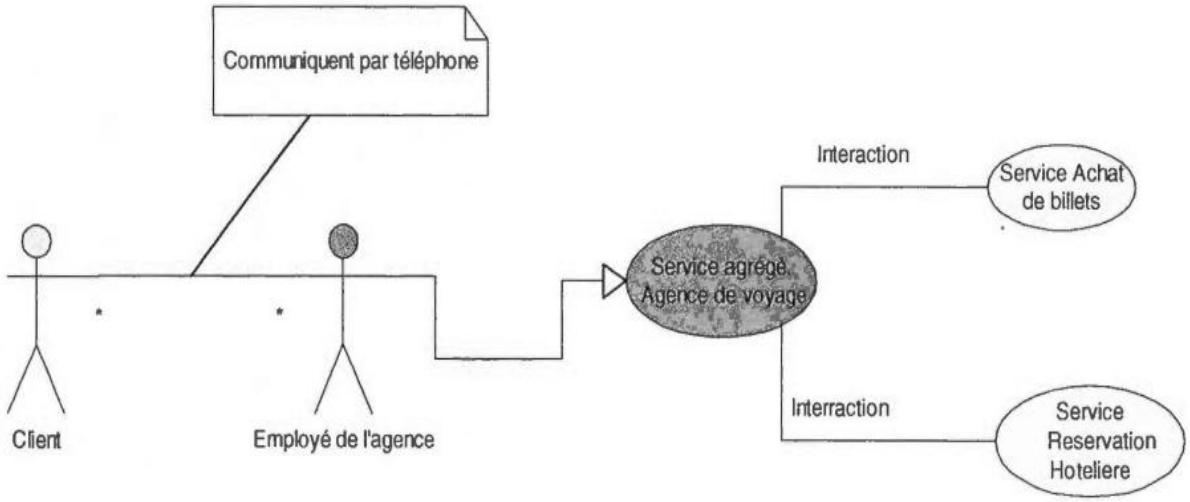


Figure 3.1

On peut voir un diagramme de cas d'utilisation pour ce dernier exemple :

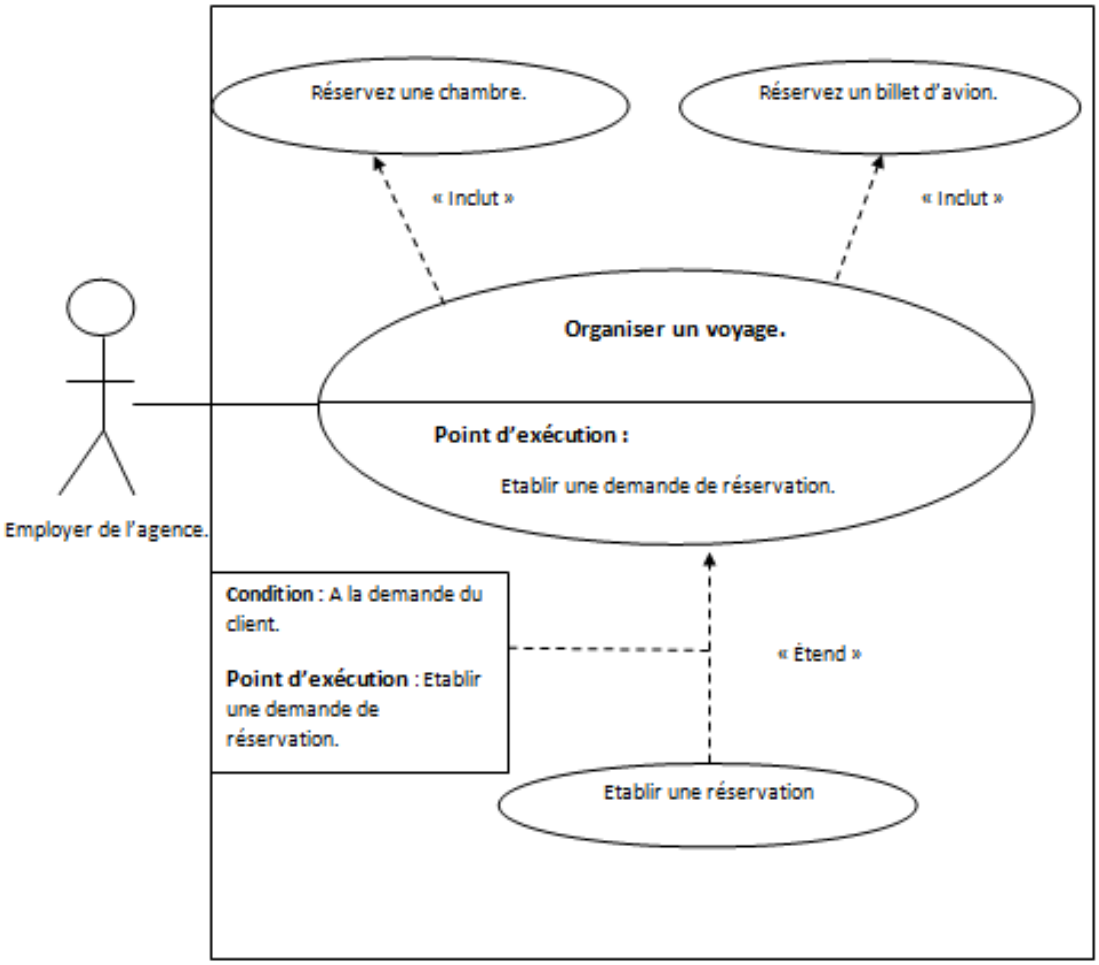


Figure3.2

On parle dans cet exemple d’une composition de services web.

Les services Web complexes peuvent à leur tour être classés en fonction de la façon dont ils sont composés des services simples. Certains services Web complexes composent des services simples présentent un comportement programmatique alors que d'autres composent des services qui présentent principalement un comportement interactif dans lequel l'entrée doit fournir par l'utilisateur. Cela rend naturel de faire la distinction entre les deux types de services Web suivants:

- Les services Web complexes qui composent les services Web programmatiques:

Les clients de ces services Web peuvent assembler des services simples, pour construire des services complexes.

- Services complexes qui composent les services Web interactifs:

Ces services exposent la fonctionnalité d'une couche de présentation de l'application Web. Ils exposent fréquemment une à plusieurs étapes le comportement de l'application Web qui combine un serveur Web, un serveur d'applications et des systèmes de bases de données sous-jacentes et typiquement fournissent une application directe sur un navigateur et, éventuellement, l'interaction d’un utilisateur humain.

Les clients de ces services Web peuvent intégrer des processus interactifs d'affaires dans leurs applications Web, des applications intégrées présentant les fournisseurs des services externes. Il est évident que des services interactifs peuvent être combinés avec les services programmatiques délivrant ainsi les processus d'affaires qui combinent la logique métier typique fonctionnalité avec le navigateur Web de l'interactivité.

2-Les modèles de composition de services Web :

Les services Web peuvent être composés en utilisant des modèles différents. Ces modèles sont sur la base des modèles de flux de travail habituels et ils sont analysés dans le présent paragraphe. Il y a de nombreuses situations où deux ou plusieurs de ces modèles sont combinés pour créer une composition complexe de service Web. Nous illustrons chacun des modèles avec un chiffre.

2.1- Modèle Séquentielle :

Le modèle séquentielle indique que les services Web sont exécutées l'une après l'autre, pas dans la séquence bouton parallèle. L'ordre séquentiel est prédéfini et doit être suivi afin d'avoir une exécution réussie.

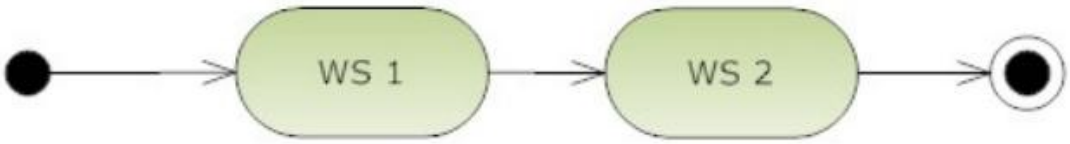


Figure 3.3

2.2-Modèle parallèle :

Le modèle parallèle (figure 3.4) indique que deux ou plusieurs services Web peuvent être exécutées en parallèle. Cela veut dire qu'ils sont indépendants et l'exécution de WS2 n'a aucune incidence sur l'exécution de WS3. L'ordre dans lequel ils sont traités n’est pas défini, mais à la fin ils sont fusionnés avec la synchronisation.

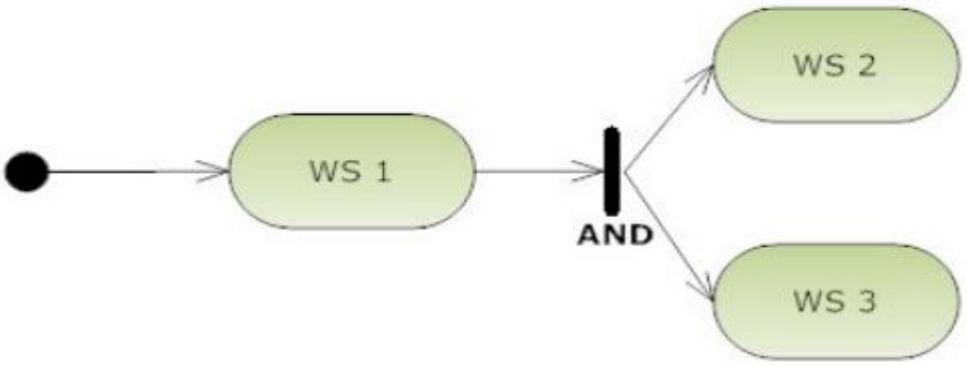


Figure 3.4

2.3-Modèle de synchronisation :

Le modèle de synchronisation (figure 3.5) indique que le processus sera poursuivre après le modèle parallèle du service Web est exécuté. Ce modèle est mis en œuvre principalement par l'utilisation de recevoir des relevés.

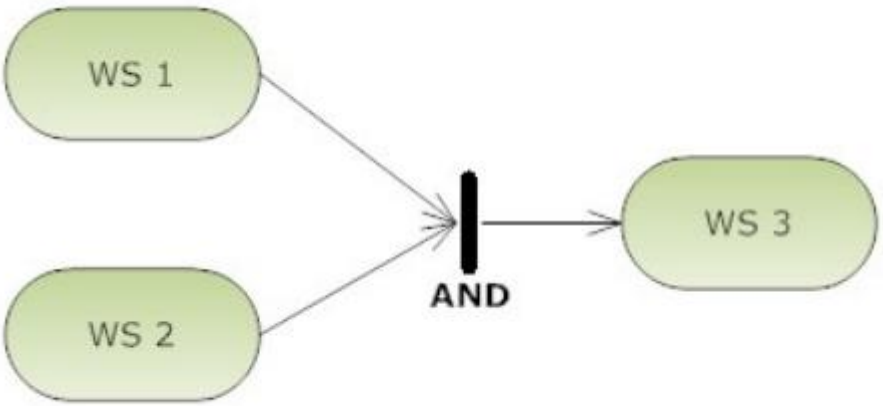


Figure3.5

2.4-Modèle de choix exclusif :

Le modèle de choix exclusif (figure 2.6) définit un point dans flux de travail de l'entreprise, où une certaine condition fondée sur une décision dans le flux est pris. En fait une condition XOR est utilisée.

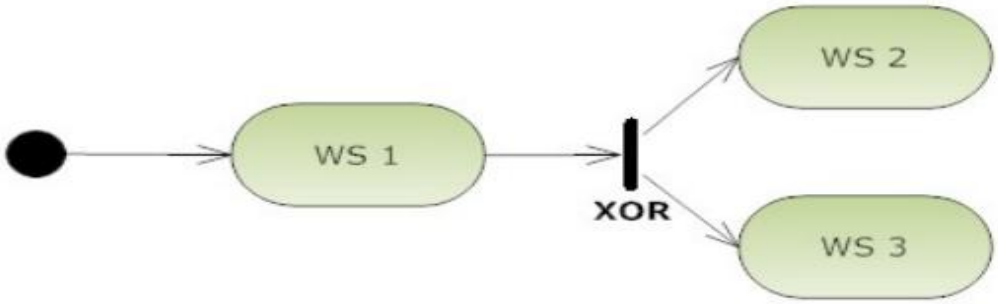


Figure3.6

2.5-Modèle de fusion simple :

Le modèle de fusion simple (figure 3.7) définit un point dans le flux d'exécution, où deux ou plusieurs branches alternatives sont fusionnées. Il est important de mentionner que le modèle de fusion simple ne prend pas en charge tout type de synchronisation, ce qui signifie, qu'aucun des procédés alternatifs n’est toujours exécuté en parallèle.

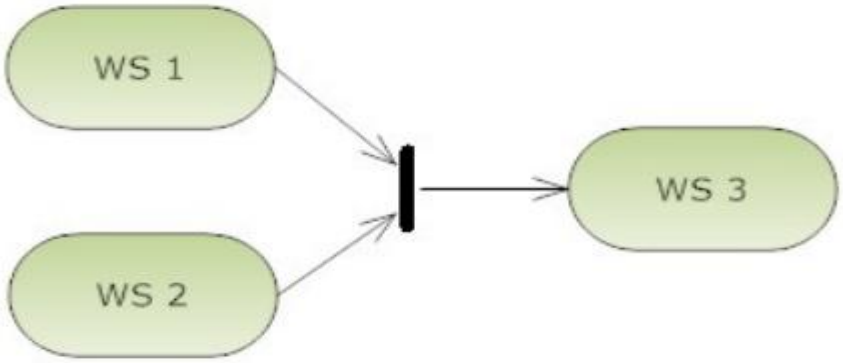


Figure3.7

2.6-Modèle de boucle :

Le modèle de la boucle (figure 3.8) indique qu'un certain point dans la composition bloc est exécuté à plusieurs reprises. Il doit n'y avoir aucune restriction sur le nombre, emplacement, et l'imbrication de ces points. Dans nos exemples, nous utilisons la boucle while.

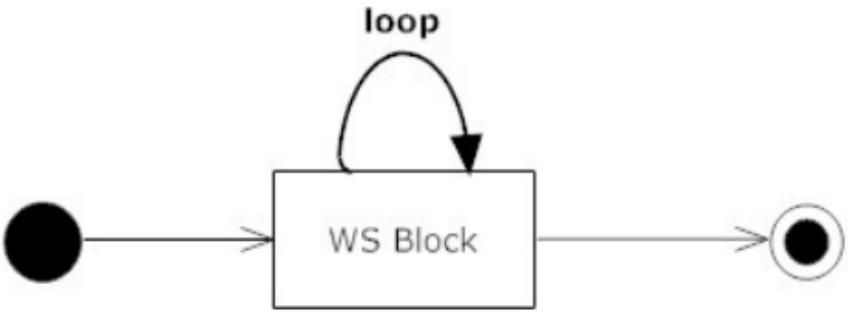


Figure 3.8

3-Calcul des paramètres de base pour les services composés :

Dans cette section, nous étudions le problème du calcul de certains paramètres de base pour les services web composés disponibles, présentant les valeurs correspondantes des constituants services simples. Comme il est difficile et inexact de définir des formules pour tous les paramètres mentionné dans le chapitre 2, nous avons fait un effort pour calculer les métriques les plus couramment utilisés pour autant que les modèles de composition possibles. Ces mesures sont les suivantes:

Le temps de réponse, le débit, la fiabilité et le coût, défini comme la température ambiante (s), T (s), R (s) et C (s) respectivement. Les services Web constitutifs sont appelés s1, s2, ..., sn et le web

Composition de service qui inclut ces services w (s1, s2, ..., sn). Pour le modèle conditionnel on note que pi la probabilité d'un service si à sélectionner.

Enfin, nous notons que SO (si, pi) l'opération de sélection pour les modèles conditionnels, qui sélectionne le service avec l'exécution probabilité pi.

3.1-Le calcule du temps de réponse :

La mesure du temps de réponse calcule le temps d'un fournisseur de services doit desservir une demande de client. Elle est généralement définie en millisecondes et est la métrique la plus couramment utilisé pour les services Web.

Pour le modèle séquentiel, le temps de réponse est défini comme étant la somme des temps de réponse des services Web constitutifs:

$$RT(s)_{sequential} = \sum_{i=1}^{n} RT(s_i)$$

Pour le parallèle, la synchronisation et le modèle de fusion simple, le temps de réponse des compositions est défini comme étant le temps de réponse maximum des services Web constitutifs.

$$RT(s)_{parallel} = \max\{RT(s_t)\}$$

$$RT(s)_{synchronization} = \max[RT(s_t)]$$

$$RT(s)_{simple\_merge} = \max\{RT(s_t)\}$$

Pour le choix exclusif, le temps de réponse est calculé par l'opération de la sélection qui sélectionne l'un des n possibles prestations de service Web. En particulier, elle est égale à la durée du service Web sélectionné de réponse est défini comme:

$$RT(s)_{exclusive\_choice} = RT(SO(s_t, p_t))$$

$$RT(s)_{deferred\_choice} = RT(SO(s_t, p_t))$$

Enfin, le temps de la configuration de la boucle de réaction est parfois le temps de réponse la boucle, où n c'est le nombre d'itérations.

$$RT(s)_{loop} = n \times RT_{(loop)}$$

3.2-Le calcule du débit :

Le débit calcule le nombre d'instances terminées par unité de temps, habituellement par seconde. Il est une mesure très utile, car il mesure la performance d'une bande d'un service. Pour les services Web composés, le débit peut être extrait si nous avons fournir les débits des services Web constitutifs.

Pour le modèle séquentiel, si nous avons des services Web n constitutifs, chacun des ont exécutés qu'une seule fois, le débit de ce service composé est défini à partir du type:

$$T(s)_{sequential} - \frac{1}{\sum_{i=1}^{n} \frac{1}{T(s_i)}}$$

Preuve: Supposons que nous ayons trois services web, avec des débits x req / s, req y / s et z REQ / s, respectivement. Ensuite, pour une invocation du premier service Web 1 / x secondes sont nécessaires. Par conséquent, 1 / y secondes et 1 / z secondes dure une invocation du deuxième et troisième service Web. Si nous composons ces services Web séquentiellement, nous avons besoin de (1 / x + 1 / y + 1 / z) secondes. Ainsi, le débit du service composé est l'inverse de cette somme.

Si nous avons des services Internet, qui sont exécutées en parallèle, le débit du service web composé est égal au minimum débit des services constituants. Cela signifie que pour que le service composé pour augmenter le débit global, il serait nécessaire d'optimiser le service avec le débit le plus bas.

Ainsi, le temps de passage, pour la parallèle, la synchronisation et la concaténation des modèles est:

$$T(s)_{parallel} = \min_i T(s_i)$$

$$T(s)_{synchronization} = \min_i T(s_i)$$

$$T(s)_{simple\_merge} = \min_i T(s_i)$$

Pour le choix exclusif et, le débit est égal au débit du service web sélectionnée :

$$T(s)_{exclusive\_choice} = T(SO(s_i, p_i))$$

Enfin, le débit d'un service web composé en utilisant le modèle de la boucle avec n itérations est défini en fonction du type:

$$T(s) = \frac{1}{\sum_{i=1}^{n} \frac{1}{T_{(loop)}}}$$

3.3-Le calcule de la fiabilité :

Comme cela est décrit dans le premier chapitre la fiabilité représente la capacité d'un service pour fonctionner correctement et de manière cohérente et est généralement exprimée en termes de nombre d'échecs transactionnels par mois ou par année. Dans notre travail, nous définissons la fiabilité en tant que la probabilité que le service peut être complété avec succès.

Pour le modèle séquentiel, parallèle, synchronisation, fusion simple et boucle, la fiabilité est définie comme étant le produit des fiabilités des services Web

constitutifs.

Cela se produit parce que, dans toutes ces situations, tous les services de la composition sont réalisés. Par exemple, si nous avons une composition de trois de services Web s1, s2, s3, suivant l'un des modèles mentionnés ci-dessus, avec une probabilité de 70%, 80% et 90% être effectuée, respectivement, la fiabilité globale du service composé est d'environ 50%.

$$R(s)_{sequentiel} = \prod_{i=1}^{n} R(s_i)$$

$$R(s)_{parallel} = \prod_{i=1}^{n} R(s_i)$$

$$R(s)_{synchronisation} = \prod_{i=1}^{n} R(s_i)$$

$$R(s)_{simple\_merge} = \prod_{i=1}^{n} R(s_i)$$

Pour le modèle de la boucle la disponibilité est égale à la disponibilité de la boucle sous tension à n, où n est le nombre d'itérations.

$$R(s)_{loop} - R_{loop}{}^{n}$$

Pour le choix exclusif, l'ensemble des fiabilités est définie par la fiabilité du

service web sélectionnée. L'opération de la sélection SO est utilisée une fois de plus pour cette raison.

$$R(s)_{exclusive\ choice} = R(SO(s_i, p_i))$$

3.4-Le calcule du coût :

Le calcul du coût est défini comme le montant d'argent payé pour le service composé. Il est évident que le coût est égal à la somme des coûts des n services Web constituants, sauf pour le cas des modèles conditionnels où il est exactement le même que le coût du service sélectionné. Ainsi, le coût séquentiel, parallèle et synchronisation est le suivant:

$$C(s) = \sum_{i=1}^{n} C(s_i)$$

Pour les modèles de synchronisation, le coût est:

$$C(s) = \sum_{i=1}^{n} C(SO(s_i, p_i))$$

Pour le modèle de la boucle, si nous avons n itérations, le coût est:

$$C(s) = n \times C_{(loop)}$$

Pour les modèles de choix exclusifs, le coût est:

$$C(s) = C(SO(s_i, p_i))$$

4-Conclusion :

Dans ce chapitre on a traité la composition de services web et le calcul de certains paramètres de base pour chaque modèle de composition. Le calcul ce fait par des formules mathématique (théorique) . Le chapitre qui ce suit présente notre travail, l'implémentation et l'environnement de travail.

Chapitre4: Implémentation

1 **Introduction :**

Ce chapitre a pour objectif majeur de présenter le produit finale. . Ce chapitre est composé de trois parties : la première partie présente Langage de composition de services web, la seconde partie présente l'environnement de développement alors que la troisième partie concerne les principales interfaces graphiques.

2 Langage de composition de services web

De nombreux industriels (tels qu'IBM ou Microsoft)et consortium (tel que le W3C) travaillent afin de mettre en œuvre un langage de composition de services Web standard (tel que WSCI –Web Service Choreography Integration)[Arkin et al.,2002]. Dans cette section, nous étudions les langages qui sont soit largement utilisés dans l'industrie (BPEL4WS), soit en cours de standardisation (WS-CDL) .

2.1BPEL4WS (Business Process Execution Language for Web Services)

BEA, IBM, SAP, Siebel Systems et Microsoft ont uni leurs efforts afin de produire un langage de composition de services Web, conçu pour supporter les processus métier à travers les services Web. Ce langage, BPEL4WS (Business Process Execution Language for Web Services), est issu de la fusion de deux langages : WSFL – Web Service Flow Language d'IBM et XLANG de Microsoft. BPEL4WS combine les caractéristiques d'un langage de processus structuré par bloc (XLANG) avec ceux d'un langage de processus basé sur les processus métier (WSFL).

BPEL4WS est basé sur XML et sur les Workflows. Ce langage distingue les processus abstraits des processus exécutables :

Le processus abstrait :ce processus spécifie les messages échangés entre les différentes parties (services Web composants) sans indiquer le comportement

de chacune d'elles. [Wynen, 2003] parle de Business Protocol, c'est-à-dire la spécification du comportement des partenaires par rapport aux messages échangés, sans rendre public le comportement interne. Ce processus abstrait peut être relié à une composition de type chorégraphie. Les services Web communiquent alors à l'aide d'échanges de messages (partie gauche de la Figure 4.1).

Le processus exécutable : ce processus permet de spécifier l'ordre d'exécution des activités, le partenaire concerné, les messages échangés entre ces partenaires, et les mécanismes des erreurs et des exceptions. En d'autres termes, il s'agit du moteur de l'orchestration donnant une représentation indépendante des interactions entre les partenaires.

La Figure 4.1 illustre la mise en œuvre des deux types de processus (exécutable et abstrait) par BPEL4WS. La définition du processus métier est donnée par le processus exécutable. Les processus abstraits gèrent les invocations entre les différents services Web permettant l'exécution de la composition de services définie dans le processus exécutable.

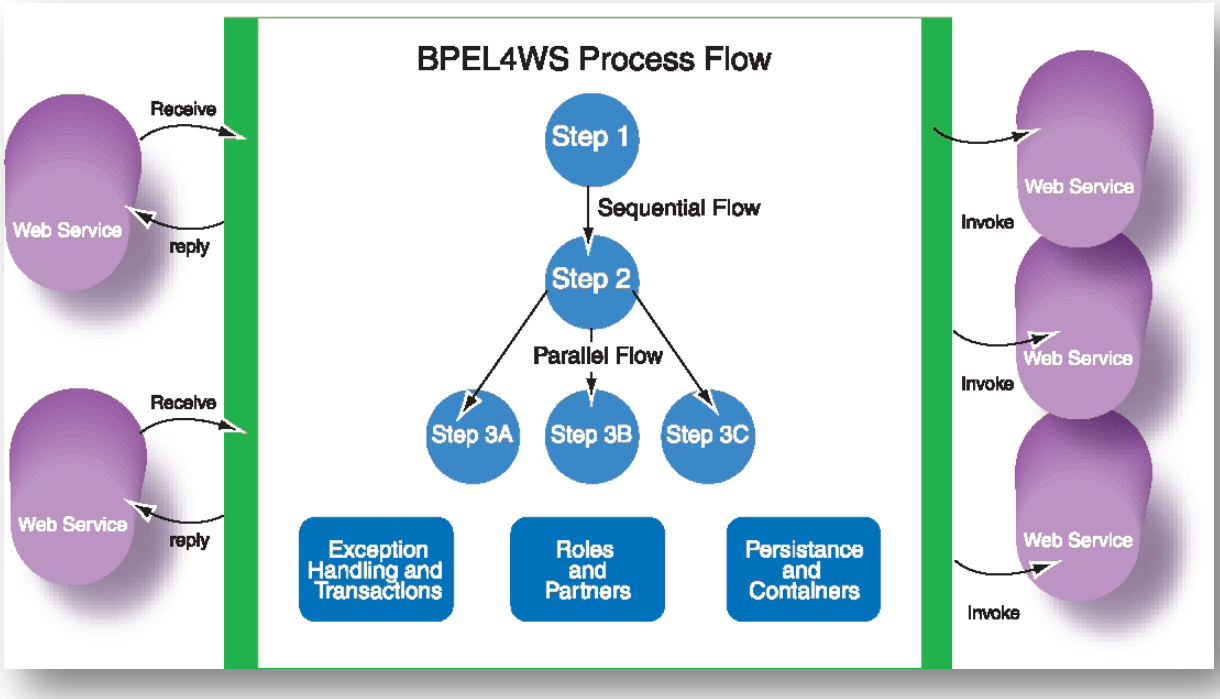


Figure 4.1 : Le flot de processus avec BPEL4WS, d'après [Peltz, 2003]

Trois éléments permettent à BPEL4WS de gérer le flot de processus dans le processus exécutable : les transactions (Exception handling and transactions), les partenaires (Roles and Partners) et les espaces de stockage (Persistance and Containers) (voir Figure 4.1) :

Les transactions. Les transactions sont utilisées dans BPEL4WS pour gérer les erreurs et les appels d'autres services si le service appelé est indisponible ou défaillant.

Les partenaires. Les partenaires sont différents services Web invoqués dans le processus. Ils ont chacun un rôle spécifique dans un processus donné. Chaque partenaire est décrit par son nom, son rôle (en tant que service indépendant), et son rôle dans le processus.

Les espaces de stockage. Les espaces de stockage permettent la transmission des données. Le flot de processus BPEL4WS permet que ces données soient cohérentes à travers les messages échangés entre les services Web. Un message peut être un message d'appel (invoke), de réponse (reply) ou d'attente (receive).

BPEL contient deux ensembles de constructeurs : constructeurs primitifs et constructeurs structurels. Ces constructeurs permettent la description d'un processus d'affaire exécutable.

Les constructeurs primitifs permettent des fonctionnalités. Parmi ces constructeurs on trouve :

- **<Invoque> :** invoque un service Web.
- **<Receive> :** attends qu'un client ou un service invoque le business process par envoie de requête.

- **<Reply> :** génère une réponse synchrone.
- **<Assign> :** permet de manipuler les données des variables.
- **<Throw> :** indique les erreurs et les exceptions.
- **<Wait> :** bloque l'exécution pour un temps donné.

Les constructeurs primitifs peuvent être combinés pour définir un algorithme complexe spécifiant les étapes par lesquelles passe le business process en utilisant les constructeurs structurels tel que :

- **<Sequence> :** définit un ensemble de constructeurs invoqués par ordre d'apparition dans la séquence
- **<Flow>** : définit un ensemble de constructeurs invoqués en parallèle
- **<Switch> :** pour le choix d'une branche à exécuter
- **<while> :** définit une boucle.

BPEL4WS est le premier langage de composition de services Web adopté par la communauté des services Web. Ceci est principalement dû au fait que ce langage possède une grande expressivité dans la définition du processus exécutable. L'inconvénient principal de BPEL4WS est que la définition du processus est rigide. Si une activité du processus échoue, le processus dans son intégralité échoue. Aucun retour en arrière et aucune alternative au processus ne sont possibles. De même, lors de la description du processus exécutable, la définition des flots de données ne permet pas de connecter des services dont les entrées/sorties ne correspondent pas exactement. BPEL4WS ne prévoit pas de mécanisme de transformation de données.

2.2WS-CDL (Web Services Choreography Description Language)

Le WS-CDL(Web Services Choreography Description Language) est un langage qui permet de décrire une vision globale des collaborations entre les services,[Kavantzas et al., 2005] insistent sur le fait que WS-CDL n'est pas

un langage de description de processus exécutables, ni un langage d'implémentation. Il décrit les séquences ordonnées de messages impliquant plusieurs entités visant à accomplir un objectif commun.WS-CDL reprend et développe la spécification Web Service Choreography Interface WSCI.

WS-CDL permet :

1. de désigner les variables et les types de données échangées,
2. de décrire les activités impliquées et
3. de décrire les structures illustrant les interactions entre les activités.

WS-CDL consiste à définir un fichier XML décrivant une chorégraphie.Les éléments d'un document WS-CDL sont illustrés en Figure 4.2.

Nous détaillons l'élément <choreography> décrivant des règles générales d'échange de messages. Il permet de définir trois aspects d'une chorégraphie:

1. Choreography Life-line décrite une collaboration et indique son état d'avancement,
2. Choreography Exception Blocks exprime les actions à entreprendre en cas d'erreur,
3. Choreography Finalizer Blocks indique les méthodes permettant de valider les résultats d'une chorégraphie ou de les annuler.

En résumé WS-CDL permet de décrire les règles selon lesquelles une collaboration doit avoir lieu. Il fournit une structuration globale de l'interaction en fonction de laquelle chaque participant décrit son processus métier et par la suite ses services.

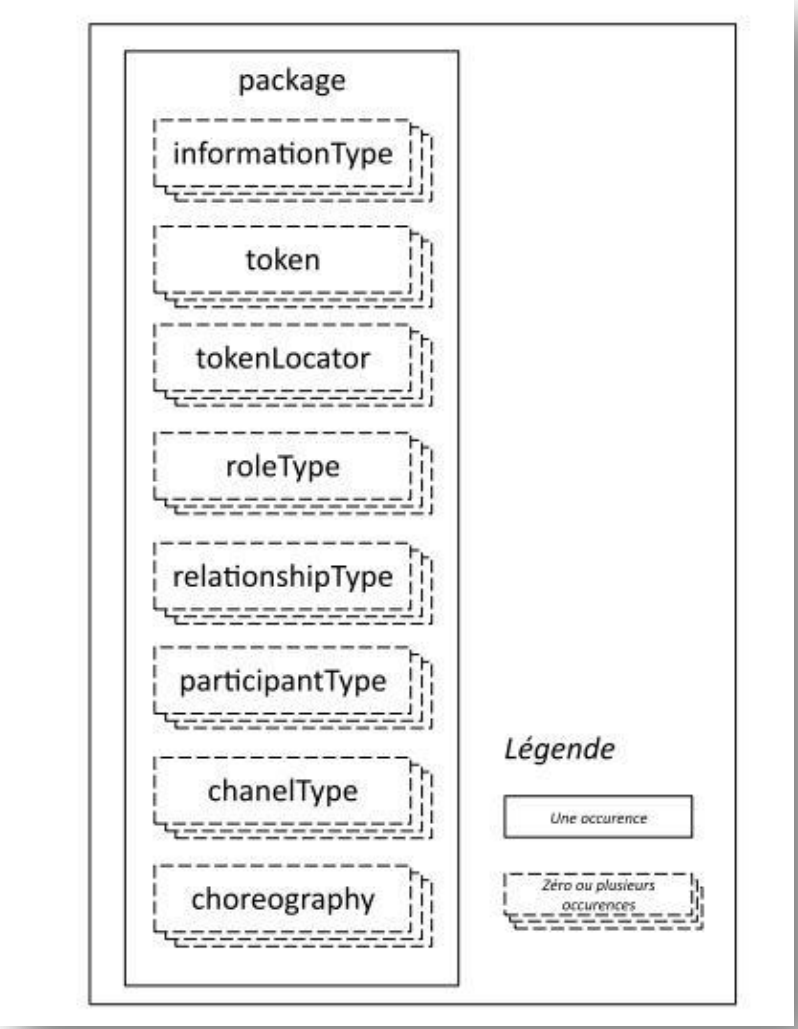


Figure 4.2: Eléments de description WS-CDL

2-Environnement de Développement

Au niveau de cette dernière partie, nous allons énumérer les outils soft que nous avons utilisés pour réaliser notre travail qui a le but de contrôler l'un des importants paramètres de Qos qui est le temps de réponse des services web composites.

Pour réaliser notre travail, nous avons opté pour les outils suivants :

- **Le langage JAVA**

Pour le langage de programmation notre choix s'est porté sur le langage JAVA, et cela parce que JAVA est un langage orienté objet simple ce qui réduit les risques d'incohérence; il est portable, il peut être utilisé sous Windows, sous Linux, sous Macintosh et sur d'autres plateformes sans aucune médication , enfin il possède une riche bibliothèque de classes comprenant des fonctions diverses telles que les fonctions standards, le système de gestion de fichiers , les fonctions multimédia et beaucoup d'autres fonctionnalités;

Bpel Designer :

L'objectif du projet BPEL est d'ajouter un support complet à eclipse pour la définition, la création, l'édition, le déploiement, les tests et le débogage des processus WS-BPEL 2.0. WS-BPEL (Web Services Business Process Execution Language), ou BPEL, est une spécification fournisseur neutre développé par OASIS pour spécifier les processus d'aires comme un ensemble d'interactions entre services Web. En fournissant ces outils.

La mise en œuvre sera extensible à des fournisseurs tiers dans un certain nombre de façons. L'éditeur sera extensible pour soutenir de nouveaux types d'activités, les pages de propriétés pour l'extensibilité de constructions existantes, une palette extensible, et les capacités de marque des produits spécifiques. Le cadre de déploiement runtime sera extensible de sorte que des tiers peuvent ajouter le support pour une variété de moteurs d'exécution. Le modèle sera en charge les extensions de fournir de nouvelles activités ou des attributs, et le validateur permettra de validation de ces extensions.

Eclipse :

Pour le choix de l'environnement de développement, on a opté pour Eclipse car il possède de nombreux points forts qui sont à l'origine de son énorme succès dont les principaux sont :

- Une plateforme ouverte pour le développement d'applications et extensible grâce à un mécanisme de plugins;
- Support de plusieurs plates-formes d'exécution : Windows, Linux, Mac OS;

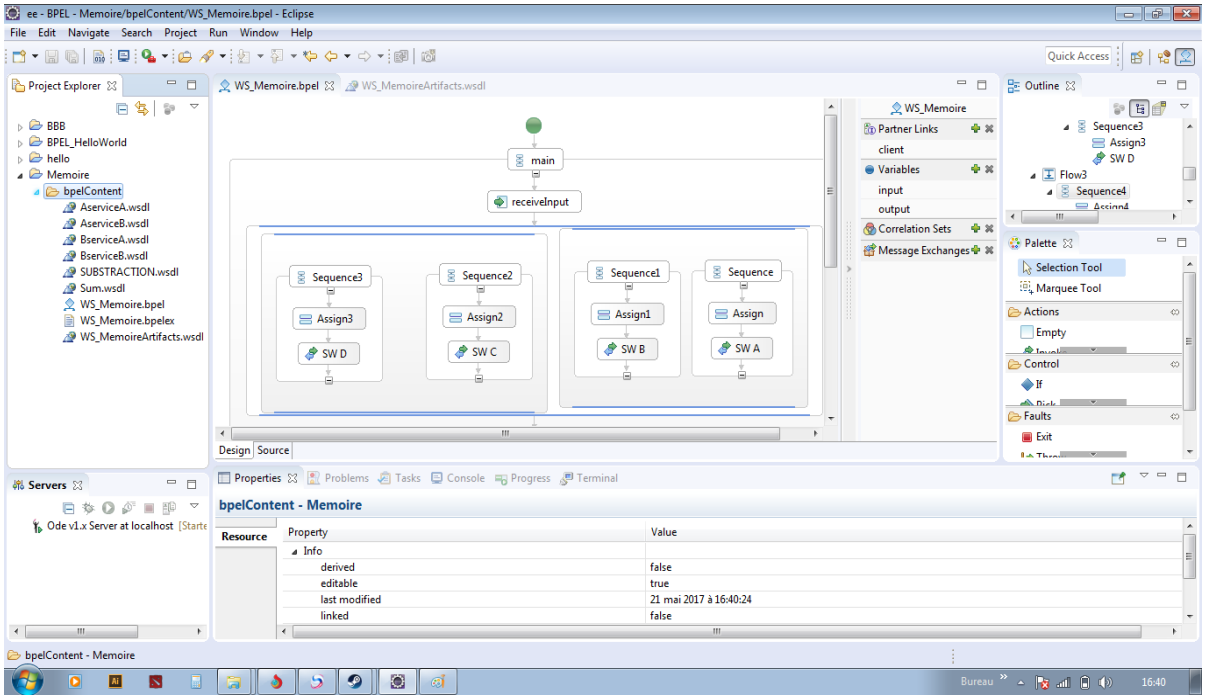
Eclipse est un environnement de développement intégré (IDE) utilisé dans la programmation. Il contient un espace de travail de base et un système de plug-in extensible pour personnaliser l'environnement. Eclipse est écrite en Java et de son utilisation principale est pour le développement d'applications Java, mais il peut également être utilisé pour développer des applications dans d'autres langages de programmation par l'utilisation de plugins, y compris: Ada, ABAP, C, C ++, COBOL, Fortran, Haskell , JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (y compris Ruby on Rails framework), Rust, Scala, Clojure, Groovy, Scheme, et Erlang. Il peut également être utilisé pour développer des packages pour le logiciel Mathematica. Les environnements de développement comprennent les outils de développement Eclipse Java (JDT) pour Java et Scala, Eclipse CDT pour C / C ++ et Eclipse PDT pour PHP, entre autres.

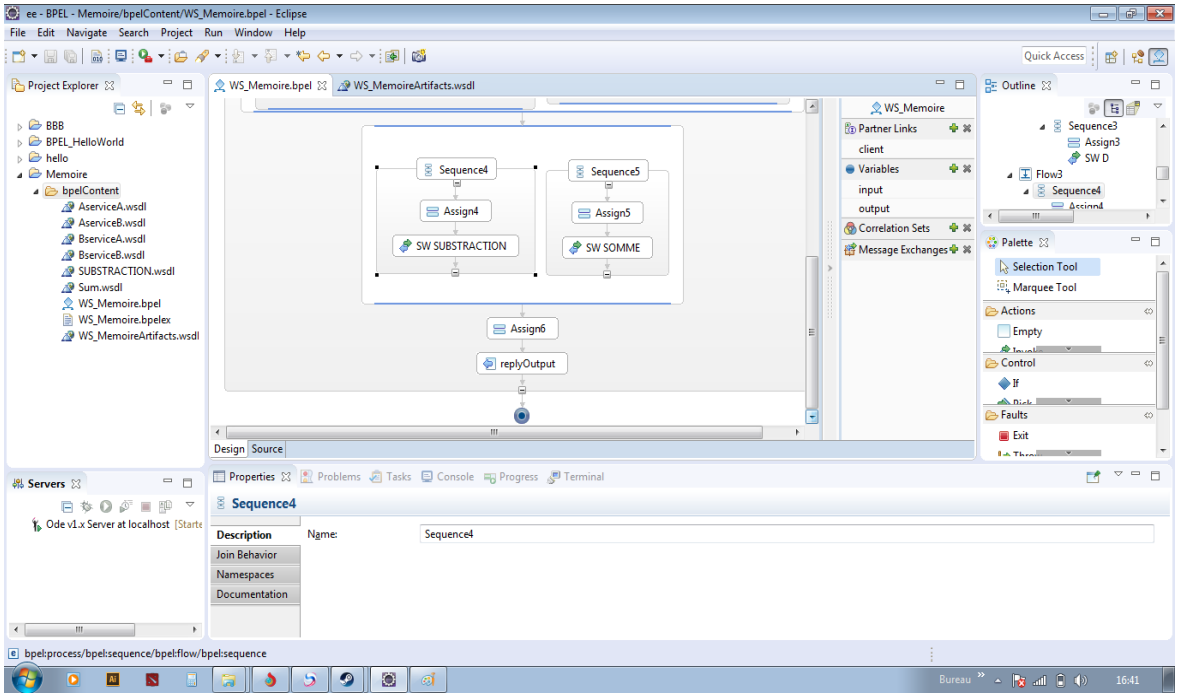
4 Présentation du travail :

Notre travail consiste a créer une application qui contrôle le Qos des services web composite, parmi ces critères non fonctionnel le temps de réponse qui est un critère très important pour les services web affin d'amélioré ces dernier. La première des chose qu'on a fait c'est la création des services web a l'aide d'eclipse puis de faire la composition en utilisant Active Bpel, de faire l'appel et enfin de contrôler le temps de réponse. On a utilisé comme modèle de

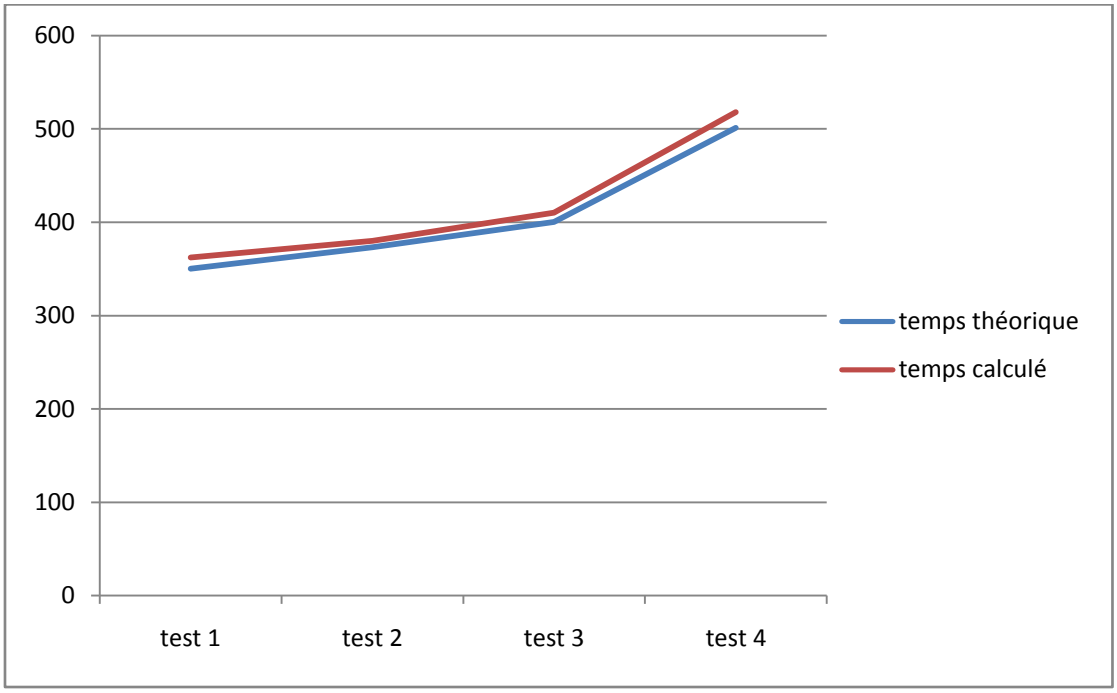
composition le modèle séquentielle et le modèle en parallèle; les formules de calcul du temps de réponse sont vu dans le chapitre précédant.

Dans ce projet on a 6 input et 3 output nous avons utilisé 2 modèles de composition : En séquentiel et en parallèle. Les figures suivantes présentent les graphes de composition des services web créé.





Dans la première séquence on a 6 services web en parallèle chaqu'un de ces services a un input et un output, pour la deuxième séquence on a 3 services web chaque service a deux inputs et un output .La première des chose on a calculer le temps de réponse des services web simple pour calculer et comparé entre le temps théorique et le temps calculer par un programme ,le temps et calculer par milli second .Les résultats sont présentées dans la figure suivante:



D’après la figure on peut voir qu'il existe une petite différence entre le temps de réponse théorique et le temps calculé.

5 Conclusion:

Dans ce chapitre nous avons présenter les différents langage de composition et l'environnement de développement, Au cours de cette dernière étape de notre travail nous avons réaliser un contrôle de temps de réponse.

Conclusion :

Notre travail est séparé en deux domaines de connaissances différents, mais liés au Services Web: contrôle de QoS des services Web et le calcul de QoS des services Web composés.

En ce qui concerne le calcul de la QoS , nous proposons une certaine métrique de formules de base, pour autant que nous sachions le modèle de composition, qui peut être une combinaison d'autres motifs. Sur cette base, nous utilisons les formules appropriées à calculer le temps de réaction, le rendement, la fiabilité et le coût des services Web composée. D'après les résultats expérimentaux, nous concluons que le types proposés sont suffisamment précis et peuvent être utilisés dans tous les services complexes composé, décrites avec les modèles mentionnés ci-dessus. Les expériences se concentrent essentiellement sur l'utilisation d'une variété de modèle de composition dans le moteur BPEL et résultats valident dans une large mesure pour les métriques de calcul pour les prestations des services composés. Enfin, nous pensons que la question principale reste pour les travaux futurs est d’examiner plusieurs formules pour toutes les mesures possibles et les modèles de composition. En outre, il serait idéal pour faire fonctionner un système de contrôle à l'exécution avec les services Web et le moteur d'exécution. De cette façon, le fournisseur pourrait prendre des résultats statistiques et correcte toute violation très rapidement et sans aucun médiateur. Une autre direction intéressante est de se concentrer sur les actions correctives après une violation est détectée.

Glossaire :

BPEL : Business Process Execution Language est un langage de processus qui dispose d'une syntaxe XML et permet la mise en œuvre de services Web complexes.

Qos: Quality of service

RPC: Remote procedure call

RST: Representational State Transfer

SLA : est une partie des accords conclus entre un consommateur de services et un fournisseur de services. Il est utilisé pour décrire de manière structuré les éléments de qualité de service (Quality of Service, QoS) contractualisés entre les deux parties.

SOA : Service orienté architecture

SOAP : Simple Object Access Protocol

XML: Extensible Markup Language

WSDL: Web Services Description Language

Bibligraphie :

Les livres :

1-Chrysostomos Zeginis, Monitoring the QoS of Web Services using SLAs - Computing metrics for composed services, University of Crete.

2- F. Barbon, P. Traverso, M. Pistore, M. Trainotti. "Run-Time Monitoring of Instances and Classes of Web Service Compositions". In Procs of ICWS'06, Salt Lake City, Utah, USA, July 2006.

3- K. Mahbub and G. Spanoudakis. "A Framework for Requirements

Monitoring of Service Based Systems". In Procs of ASE'04, Linz, Austria, September 2004.

4- Monitoring conversational web services.

5- Qos issues in web services

6- Runtime monitoring in service-oriented architecture.

7- Smart monitors for composed services.

8-Samir Youcef, Extension de l'architecture conventionnelle des services web(SW) :prise en compte de la qualité de service dans la sélection des SW, Laboratoire d'analyser et de modélisation des systèmes pour l'aide à la décision 2, Place du Maréchal de Lattre de Tassigny, BP 75775 cedex 16, France.

9-Turki Hazar, L'étude de cas pour la sélection des services web basée sur les contraintes temporelles, conférence février 2009.

9- Zhang W., Yang Y., Tang S., L., « Fang. Qos-driven service selection optimization model and algorithms for composite web services. », 31st Annual International Computer Software and Applications Conference (COMPSAC'2007), IEEE Computer Society, 2007.

Les sites:

http://akrambenaissi.com/2010/06/04/historique-des services web

Table de matière