



**République Algérienne Démocratique et
Populaire**

**Ministère de l'Enseignement Supérieur et de la
Recherche Scientifique**

**UNIVERSITÉ ECHAHID HAMMA LAKHDAR
EL OUED**

FACULTÉ DES SCIENCES ET DE TECHNOLOGIE

Mémoire de fin d'étude

LICENCE ACADEMIQUE

Domaine: Mathématiques et Informatique

Filière: Informatique

Spécialité: Informatique

Thème

Intitulé du mémoire

**Implémentation d'un processus métier en utilisant la
composition statique de services Web**

**Application : Traitement d'un appel d'accident en utilisant
BPEL**

Présenté par: Baraika Imane
Hani Messaouda

Sous la supervision de :

MEKLID ABDESALAM

Année universitaire 2014 – 2015

Dédicaces

Avant tout, je remercie dieu le tout puissant de m'avoir donné le courage et la patience pour réaliser ce travail malgré toutes les difficultés rencontrées.

Je dédie ce modeste travail

À mes très chers parents adorés qui m'ont aidé, grâce à leur prière et à leur

bénédiction, à avoir la confiance et l'espoir au cours de la rédaction de ce travail.

Je le dédie aussi À mes très chers frères Sadok , Med Dhia .

À ma très chères soeur Amel et ses enfant "Aroua" et "Ritadje"

Et à toute la famille.

À mes intimes amies Maroua, Houda , Chaïma, Zineb , Imane

À tous les étudiants de la faculté Informatique surtout les étudiants de

la 3ème année promotion 2015.....

Messouda



Dédicaces

Avant tout, je remercie dieu le tout puissant de m'avoir donné le courage et la patience pour réaliser ce travail malgré toutes les difficultés rencontrées.

Je dédie ce modeste travail

À mes très chers parents adorés qui m'ont aidé, grâce à leur prière et à leur

bénédiction, à avoir la confiance et l'espoir au cours de la rédaction de ce travail.

Je le dédie aussi À mes très chers frères Djmouí .

À ma très chères sœur Massouda et ses enfant "Farída" et

"Uamina" et "Ibrahim"

Et à toute la famille.

*À mes intimes amies Massouda , Nasira , Assma , Zohra ,
Salha*

*À tous les étudiants de la faculté Informatique surtout les
étudiants de*

la 3éme année promotion 2015.....

Imane



Remerciements

Nous remercions « Notre Dieu Allah avant tout » qui nous donne la puissance pour compléter ce travail,

Nous adressons nos vifs remerciements à notre encadreur MEKLID ABDESALAM qui nous aidés durant le travail et leur patience et sa valeur conseils

Mes remerciements s'adressent également aux tous les profs de l'université d'El-oued en particulier BOUZAHER ABDELAZIZ qui nous donne aide inestimable

On n'oublie pas l'aide de nous camarades CHaima hafouda et khicha yahia

Et enfin , je tiens à remercier tous ceux qui ont contribué, de près ou de loin, par leurs encouragements, leurs conseils et leurs soutiens à mener à ce travail.



:

يعاني الكثير من الاشخاص على الطرقات من الحوادث و خاصة الطرق المعزولة و البعيدة عن المدن.

و يتلخص مشروعنا في محاولة انجاز برنامج خاص بالحوادث يعتمد على تركيب مجموعة من الويب في تسهيل الابلاغ عن الحوادث بطريقة مثالية كما يسمح بإرسال الحماية و الشرطة الى موقع

.

خدمات ويب

الكلمات المفتاحية :

Sommaire

Liste des figures

Abbreviations

Introduction générale01

Partie I : Contexte de travail

Introduction03

1. service web04

1.1. Définition de service web.....04

1.2. Caractéristiques Des Services Web.....04

1.3. Avantages05

1.4. Inconvénients.....06

1.5. Architecture des services Web.....06

2. composition de service web09

2.1. Définition composition de service web.....09

2.2. Type de composition de service web09

2.3. Techniques de composition de services web11

Conclusion13

Partie II : modélisation

Introduction14

1. Objectifs15

2. Modélisation du système15

2.1. Diagramme de cas d'utilisateur	15
2.2. Diagramme de séquence	16
2.3. Diagramme d'activité du processus	17
3. Fonctionnement du système d'appel d'accident	17
4. Définition des Services Web du Système	18
4.1. Service accident	18
4.2. Service Pompier_ Hôpital	19
4.3. Service 17	19
Conclusion	20
Partie III : implémentation	
Introduction	21
1. Les Outils utilisées	22
2. Déploiement de service composite	24
2.1. Test le web service	24
2.2. Design de BPEL	25
2.3. Composition de service	26
2.4. Test de l'application de composition (Composite Application)	27
2.5. Exécution de L'application	27
Conclusion	30
Conclusion générale	31
Bibliographie	32

Figures et tableaux

Liste de Figures

Figure I-1 - Evolution des architectures logicielles.

Figure I-2 - Les interactions entre les services Web.

Figure I-3: Web Service Orchestration

Figure I-4: Web Services chorégraphie

Figure II-1: diagramme de cas d'utilisation

Figure II-2: Diagramme de séquence

Figure II-3 : Diagramme de activité de processus

Figure II-4 : Fonctionnement du système

Figure III-1: test le service « accident » .

Figure III-2: test le service « service 14 ».

Figure III-3: test le service « ser_hopitale ».

Figure III-4: test le service « service 17 » .

Figure III-5: PBEL Module .

Figure III-6: composite application .

Figure III-7: test le composite application « requiest » .

Figure III-8: test le composite application «réponde » .

Figure III-9: interface de le utilisateur

Figure III-10: interface de saisir le cordonnée

Figure III-11 :résultat si l'accident est en nouvelle déclaration

Figure III-12: résultat si l'accident est déjà déclarée

Liste de tableaux :

Tableaux II-1 : tableaux de accident dans la base de donnée .

Tableaux II-2 : tableaux de hôpital dans la base de donnée .

Abbreviations

W3C	World Wide Web Consortium
WSDL	Service web Description Language
SOAP	Simple Object Access Protocol
HTTP	HyperText Transfer Protocol
XML	eXtensible Markup Language
URI	Uniform Resource Identifier
SMTP	Simple Mail Transfer Protocol
RMI	Remote Method Invocation
CORBA	Common Object Request Broker Architecture
DCOM	Distributed Component Object Model
BPEL	Business Process Execution Language
SGBD	Système de Gestion de Bases de Données
Mysql	Structured Query Language
UML	Unified Modelling Language

Introduction générale

D'autant que toutes les informations techniques, économiques, pratiques et culturelles sont accessibles en temps réel. L'avantage d'Internet réside dans la possibilité de pouvoir agir ou réagir beaucoup plus efficacement, les plus importants services dans les services de l'internet c'est les services web. Un service web (ou service de la toile) est un programme informatique de la famille des technologies web permettant la communication et l'échange de données et fournis des services 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, de manière synchrone ou asynchrone.

Le protocole de communication est défini dans le cadre de la norme SOAP dans la signature du service exposé (WSDL). Actuellement, le protocole de transport est essentiellement HTTP(S). Le monde industriel utilise une approche statique ce service (service web) basée sur les processus métiers afin de réaliser cet enchainement, Une autre approche, consiste à composer les services dynamiquement en fonction de leurs fonctionnalités et de leurs disponibilités.

Depuis quelques années, de nombreuses entreprises adoptent la technologie des services web pour développer leurs activités sur le net. Pareillement, les développeurs intègrent de plus en plus les services web dans leurs applications. Ce pendant, certains services ne peuvent seuls satisfaire les besoins d'un client. Une composition des services disponibles est donc nécessaire. L'idée de la composition des services web est de créer un nouveau service en combinant des services existants en se basant sur leurs comportements pour répondre aux besoins de l'utilisateur, C'est dans ce contexte que notre projet vise à apporter une composition des services web.

Notre objectif est l'implémentation du processus métier de traitement d'appel d'urgence en utilisant la composition statique des services web avec le langage BPEL

La première partie de notre travail survole le contexte de notre travail ; introduira le concept de base des services web ; une étude bibliographique des différents types, techniques, approches et langages pour composer des services web statiquement. En suite, dans la deuxième partie , nous présenterons une modélisation de notre objet d'étude de cas. Ainsi que le langage BPEL pour la composition des services web existants. En fin nous avons montré

l'implémentation de notre Système. La démarche détaillée de notre implémentation, qui contient les différentes étapes de création de services Web et leurs orchestration

Nous avons terminé notre rapport par une conclusion générale qui contient le bilan de notre travail et quelques perspectives.



contexte de travail

Introduction

L'accès aux systèmes d'information s'appuie aujourd'hui de plus en plus sur des technologies internet. Les efforts de standardisation dans ce contexte ont accentué l'engouement des personnes et des organisations (aussi bien académiques, qu'industrielles, commerciales, ou institutionnelles) pour l'utilisation de l'Internet et ont permis l'émergence des services Web comme support de développement des applications accessibles par Internet. Ainsi, les technologies associées aux services Web sont devenues incontournables pour le développement d'applications interagissant les unes avec les autres par le biais de l'Internet. Nous proposons dans ce chapitre de faire un point sur le sujet des services Web.

1. service web

1.1. Definition de service web

Definition : Le consortium W3C (World Wide Web Consortium) définit un service Web comme étant: «A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. » [1]: - Il est identifié par un URI (Uniform Resource Identifier) [2]. - Ses interfaces et ses liens peuvent être décrits en XML. - Sa définition peut être découverte par d'autres services Web. - Il peut interagir directement avec d'autres services -Web à travers le langage XML en utilisant des protocoles Internet standards.

1.2. Caractéristiques Des Services Web

Plusieurs acteurs définissent les services web par des caractéristiques technologiques distinctives les plus importantes sont : [3]

- **Autonomie** : Les services Web sont des composants applicatifs autonomes dont le développement, le déploiement, le fonctionnement, l'administration et la sécurité varient indépendamment des clients des services.
- **Transparence** : Tous les détails de l'implémentation restent privés au sein d'un service. La façade orientée objet que chaque service expose assure l'isolation entre l'extérieur du service et les choix d'implémentation effectués par le développeur de ce service.
- **Couplage faible** : le service et son consommateur (le programme qui l'invoque) sont indépendants l'un de l'autre. Si une modification est à faire sur le consommateur, on n'a pas besoin de connaître la machine, le langage de programmation, le système d'exploitation ou autres paramètres, afin d'établir à nouveau une communication entre le service web et son consommateur. Le consommateur possède une fonctionnalité qui consiste à faire une localisation et une invocation du service web.
- **Un service web est une application logicielle qui est reconnue par un URI** : le service web est accessible en spécifiant son URI, c'est-à-dire que le service web est caractérisé par un seul objet et une seule fonctionnalité, c'est un prolongement de la

programmation orientée objet et à partir de cela, on peut faire la construction d'une application logicielle très large comportant plusieurs fonctionnalités, afin de sélectionner les fonctionnalités qui sont recherchées par les URI spécifiques.

- Capacité des interfaces et liaisons d'être publiées, localisées et invoquées via le langage XML.

- **Capacité d'interagir avec les composants des logiciels via des éléments XML avec l'utilisation des protocoles Internet standard :** L'interopérabilité est basée sur

L'utilisation du XML et des protocoles Internet standards, tels que, le HTTP qui est le protocole du web, le SMTP qui est le protocole du courrier électronique,...etc.

1.3. Avantages :

- Les services web peuvent être implémentés sur différentes plates-formes et avec des Langages variés. Ils facilitent alors l'interopérabilité entre systèmes et plates-formes Hétérogènes [4].
- Les services Web utilisent des standards et protocoles ouverts [4].
- Les services web permettent également de profiter de différents environnements et langages de développement par une publication, localisation, description et une invocation via XML. Donc les services web sont très flexibles et présentent l'avantage d'être faiblement couplés et réutilisables [4].
- Le coût d'accès aux services web est très inférieur à celui des technologies précédentes. De plus que les services web permettent d'automatiser facilement les processus métiers intra et interentreprises en échangeant des informations au format XML [4].
- Les outils de développement, s'appuyant sur ces standards, permettent la création automatique de programmes utilisant les services Web existants [3].
- Les protocoles et les formats de données sont au format texte, facilitant ainsi la compréhension du fonctionnement global des échanges [3].
- Les services web sont grandement utilisés par les entreprises, ce qui leur permet d'exposer un certain nombre de services et d'échanger les informations entre elles [3].

- Basés sur le protocole HTTP, les services Web peuvent fonctionner au travers de nombreux pare-feu (firewalls) sans nécessiter des changements sur les règles de filtrage [3].

1.4. Inconvénients :

- Les normes des services Web dans certains domaines sont actuellement récentes donc ils ne sont pas encore stable [5].
- **Disponibilité** : les services web peuvent bien satisfaire un ou plusieurs besoins de client. seront-ils pour autant toujours disponible et utilisable ? ça reste un défi pour les services web [3].
- **Performance** : Les services Web sont encore relativement faibles par rapport à d'autres approches de l'informatique répartie telles que : le RMI, CORBA, ou DCOM. [3].
- **Fiabilité** : il est difficile de s'assurer de la fiabilité d'un service car on ne peut garantir que ses fournisseurs ainsi que les personnes qu'ils invoquent travaillent d'une façon fiable [5].
- **Sécurité** : Il est facile de contourner les mesures de sécurité mises en place par les pare-feu, l'utilisation du protocole http (tel que mentionné ci-haut) n'a pas que des avantages car les normes de sécurité des services web laissent encore à désirer. CORBA par exemple est la technologie la plus mûre et la plus sécuritaire [6].
- **la lenteur de ces échanges** : vient de.
- l'utilisation du réseau. Cela signifie que si une partie importante du code d'une application dépend d'une requête à un service Web, et que le réseau, pour une raison quel conque, est paralysé, l'application sera dans l'incapacité de fonctionner correctement. Il y a donc une très forte dépendance de contraintes extérieures pas forcément contrôlables [6].
- les services Web ne sont pas encore très répandus ; il est donc encore peu probable de trouver l'objet de sa recherche [6].

1.5. Architecture des services Web

Besoin en architecture : Les architectures en informatique proposent des schémas directifs et une vision sur le fonctionnement et la dynamique des systèmes. Pour promouvoir l'interopérabilité et l'extensibilité du paradigme des services Web, une architecture de référence est nécessaire afin de préserver les objectifs initiaux visés par les services Web lors des évolutions technologiques successives.

Une architecture orientée services SOA : (Service Oriented Architecture) : L'architecture des services Web comme l'architecture du Web sont des instances d'architectures orientées services(SOA) [7]. Elle propose une perspective globale sur le développement, la gestion et le fonctionnement des services Web [8]. L'architecture SOA est un modèle (abstrait) qui définit un système par un ensemble d'agents logiciels distribués qui fonctionnent de concert afin de réaliser une fonctionnalité globale préalablement établie [9].

Elle consiste à diviser le logiciel répondant à un problème, en un ensemble d'entités proposant des services. Chacune de ces entités peut utiliser les services proposés par d'autres entités. On obtient ainsi un réseau de services interagissant entre eux. Cette architecture s'appuie sur une architecture à composants (implémentation « réelle » des services [10]) et suit l'évolution logique des architectures logicielles [11] (figure 1) :

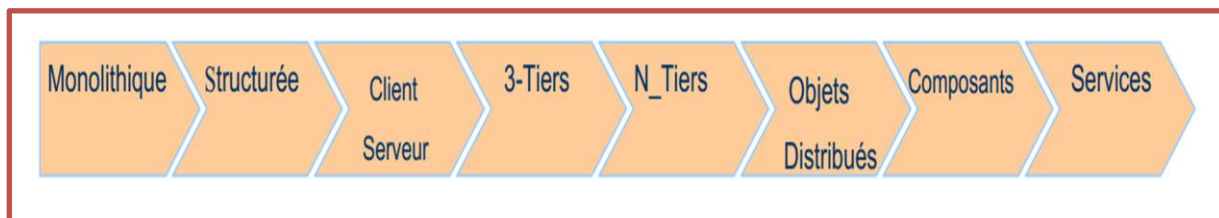


Figure I-1 - Evolution des architectures logicielles.

Le choix d'une architecture SOA entre dans la perspective de transformer le Web en une énorme plate-forme de composants faiblement couplés et automatiquement intégrables.

L'architecture SOA vise trois objectifs importants [12] : (i) identification des composants fonctionnels, (ii) définition des relations entre ces composants et (iii) établissement d'un ensemble de contraintes sur chaque composant de manière à garantir les propriétés globales de l'architecture.

Modèle de fonctionnement

Les interactions entre les services Web impliquent trois participants (acteurs) [13]: le fournisseur de services (prestataire), l'annuaire de services et le client (utilisateur du service)(voir figure2).

- Le fournisseur de services correspond au propriétaire du service (i.e. entité responsable du services Web). D'un point de vue technique, il est constitué par la plate-forme d'accueil du service.

- Le client correspond au demandeur du service. D'un point de vue technique, il est

constitué par l'application de recherche et d'invocation d'un service. L'application client peut être elle-même un service Web.

- L'annuaire des services représente l'entité logicielle qui joue le rôle de l'intermédiaire entre les clients et les fournisseurs de services, il correspond à un registre de descriptions de services offrant des facilités de publication de services pour les fournisseurs ainsi que des facilités de recherche pour les clients et le moyen de localiser leurs besoin en terme de services. La dynamique de l'architecture se décompose ainsi : Le fournisseur de services définit la description de son service et la publie dans un annuaire de service (dans des registres) en vue d'être localisé par des clients; Le client utilise les facilités de recherche disponibles au niveau de l'annuaire pour retrouver et sélectionner un service donné. Il examine ensuite la description du service sélectionné (extrait sa description du registre) pour récupérer les informations nécessaires lui permettant de se connecter au fournisseur du service et d'interagir avec l'implémentation du service considéré (entreprend une interaction) [14]

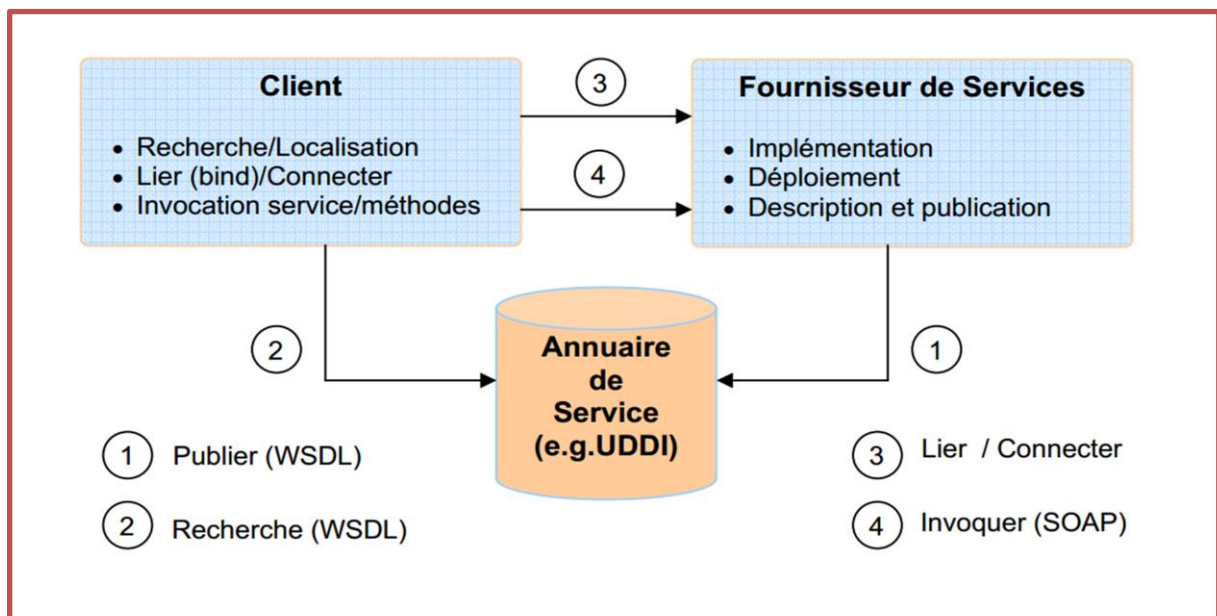


Figure I-2 - Les interactions entre les services Web.

Le modèle de fonctionnement de l'architecture des services Web se base sur un cadre technologique qui constitue l'infrastructure de l'architecture par composants des services Web.

Cette infrastructure offre les services nécessaires pour la réalisation des différentes étapes du cycle de vie d'un service Web.

2. composition de service web

2.1. Définition composition de service web

La composition de services constitue l'un des champs de recherche les plus actifs dans le domaine des services web. Nous avons trouvé plusieurs définitions pour la composition de services pratiquement toutes identiques, parmi lesquelles nous avons retenues:

« La composition de services web désigne la création d'un nouveau service en réutilisant et en combinant des services existants. La composition définit un procédé exécutable, externalisé comme un nouveau service, dont les activités composantes sont d'autres services web » [14].

En effet, la composition de services web est essentielle et plus avantageuse pour l'utilisateur. Elle prend en charge la situation où la requête d'un client ne peut être satisfaite par aucun service disponible, mais par un service composite obtenu en combinant convenablement des services existants [14]. Par conséquent, les services web composites doivent être considérés comme des services web et doivent garder les mêmes caractéristiques que les services web basiques à savoir : auto-descriptifs, interopérables, et facilement intégrables [4].

Une composition de services web spécifie quels services ont besoin d'être exécutés, dans quel ordre et comment gérer les conditions d'exception [4]. Cependant, elle peut être à plusieurs niveaux en permettant à des services web élaborés d'être à leur tour combinés pour construire de nouveaux services [14]. Les services composants existent au préalable et peuvent ne pas être fournis par la même organisation [4].

Ce que nous pouvons retenir est que la composition de services web permet de fournir un nouveau service plus complexe capable de répondre aux besoins de l'utilisateur en combinant de façon adéquate les services disponibles.

2.2. Type de composition de service web

Nous avons déjà expliqué qu'est ce qu'un service Web et comment fonctionnent les services web composites. Mais, il nous faudra encore comprendre comment cette composition peut s'accomplir depuis la perspective du degré d'automatisation. La composition peut être classifiée dans trois catégories [15]:

2.2.1. Composition 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. Composition semi-automatique

Les techniques de composition semi-automatiques sont une 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. Composition 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 rejeté.

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 notre cas.

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.

2.3. Techniques de composition de services web

La composition des Services Web peut se faire de deux manières: orchestration et chorégraphie [16] [17].

2.3.1. Orchestration

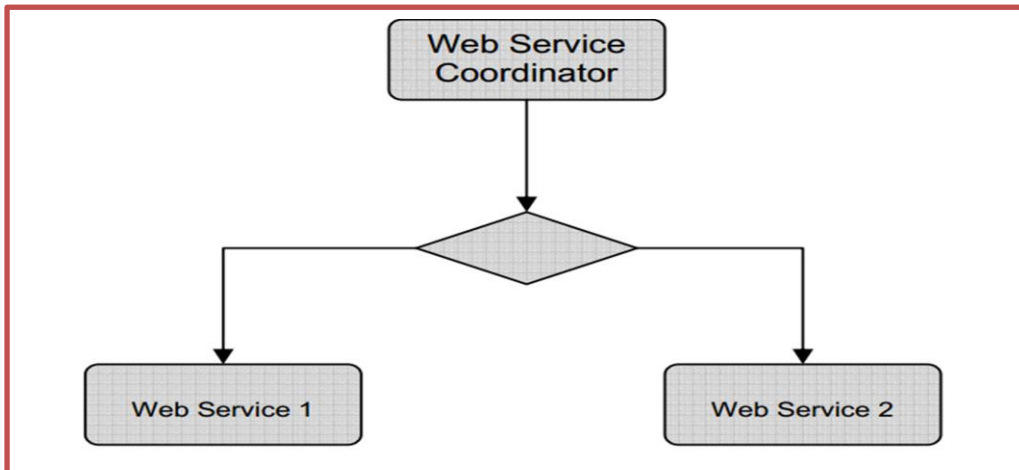


Figure I-3: Web Service Orchestration

2.3.2. Chorégraphie

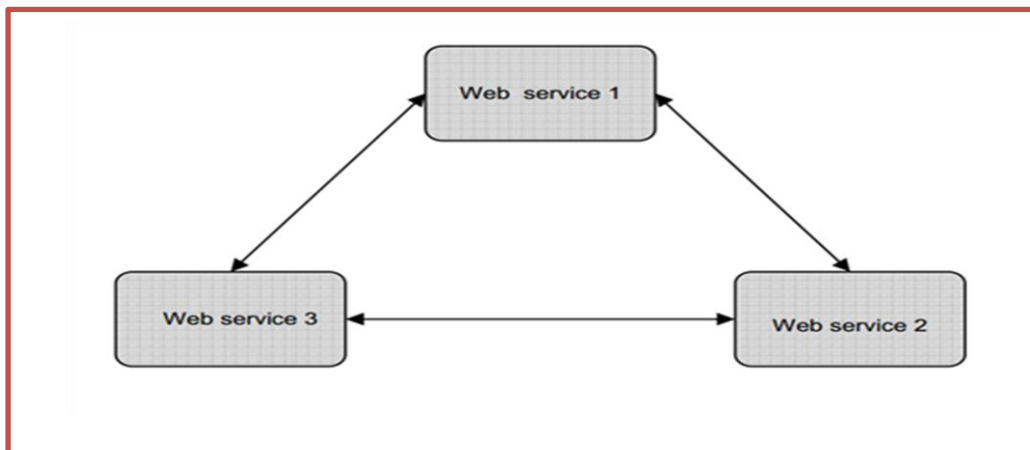


Figure I-4: Web Services chorégraphie

Depuis la perspective de la composition des services web, l'orchestration est un rapprochement plus flexible que la chorégraphie:

- Le responsable ou coordinateur de tout le processus métier est connu.
- Les services web peuvent être incorporés sans soucis, parce que ils n'ont pas conscience d'appartenir à un processus métier.

On va Choisit un technique de composition de l'Orchestration

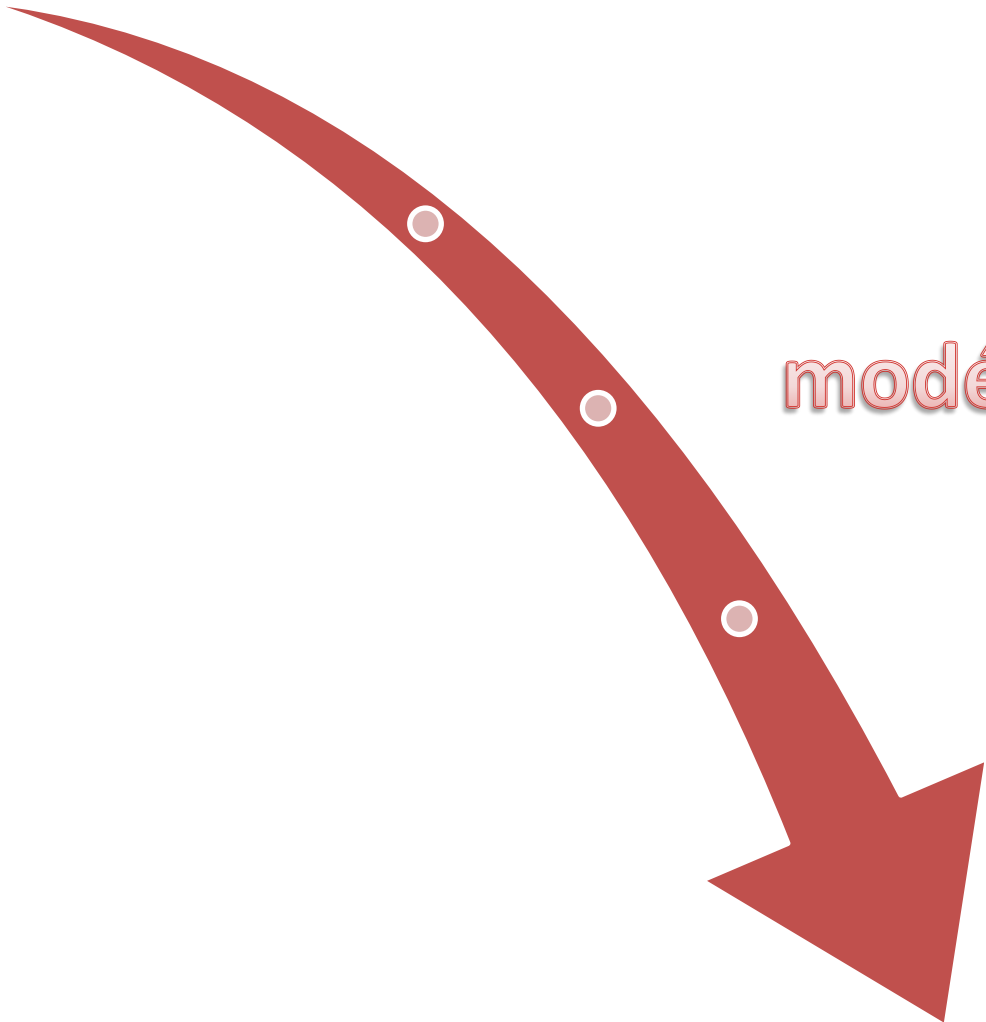
Définition : L'orchestration décrit l'interaction des services au niveau de messages, incluant le logique métier et l'ordre d'exécution des interactions. Les services web n'ont pas de connaissance (et n'ont pas besoin de l'avoir) d'être mêlées dans une composition et d'être partie d'un processus métier. Seulement le coordinateur de l'orchestration a besoin de cette connaissance.

La Figure I-3 montre le workflow dans l'orchestration des services web. Un coordinateur prend le control de tous les services web impliqués et coordonne l'exécution des différentes opérations des services web qui participent dans le processus.

Conclusion

Ce parti nous a permis de comprendre ce qu'est la composition de services web et de mesurer son importance dans le cas où la requête d'un client ne peut être satisfaite par un seul service web mais en combinant plusieurs services web.

Le but de la composition est de réutiliser des services existants pour réaliser des fonctionnalités complexes, elle permet de réduire le coût et le temps de développement des applications service web.



modélisation

Introduction

Les services web constituent un moyen puissant pour exposer les fonctionnalités sur Internet. Depuis quelques années, de nombreuses entreprises adoptent la technologie des services web pour développer leurs activités sur le net. Pareillement, les développeurs intègrent de plus en plus les services web dans leurs applications. Ce pendant, certains services ne peuvent seuls satisfaire les besoins d'un client. Une composition des services disponibles est donc nécessaire. C'est dans ce contexte que notre projet vise à apporter une composition des services web.

Dans ce qui suit, nous présenterons la conception de notre système en mettant en application les notions présentées dans le contexte de travail. Nous commencerons par une présentation de notre approche pour la composition de services web et une description de l'architecture générale du système. Nous procéderons par détailler le fonctionnement de la composition et les différents services du système de composition de services web en utilisons modelisation par les differents diagramme de representations d'UML.

1. Objectifs

Dans le cadre de notre problématique générale de composition de services web et selon notre étude de cas. L'objectif de notre approche est d'aboutir aux finalités suivantes :

- ❖ Alliances entre organismes pour offrir un service intégré à valeur ajoutée en combinant des services existants ;
- ❖ Développement d'applications distribuées de service web ;
- ❖ Développement d'un service web composite « appel accident » ;

Plus précisément, nous concevons notre solution comme un outil qui permettra à un utilisateur d'utiliser le service web composite « Appel accident », pour prendre les mesures nécessaires en cas d'accident. Pour cela, nous adopterons une approche *automatique*, car les services web seront invoquer automatiquement par le service web composite sans intervention de l'utilisateur : l'utilisateur qui n'a que saisir les coordonnées de l'accident pour que le processus continue son déroulement automatiquement sans aucune intervention.

Nous avons utilisés une composition statique. Car les services sélectionnés sont connu d'avance (à la phase de conception).

Nous utiliserons la technique de l'orchestration des services web qui permet à un coordinateur Central (Service Appel accident) de contrôler et de gérer les autres services. L'orchestration sera développée en définissant par exemple : l'ordre d'exécution des services web (séquentiel, parallèle, etc.), les entrées/sorties des services web, etc.

Nous utiliserons un langage standard pour décrire le Processus. Ce dernier est le **langage BPEL** qui nous a particulièrement intéressés grâce aux avantages qu'il apporte.

Notre travail vise à apporter des améliorations concrètes dans le domaine de secoure à distance en assurant plus de rapidité dans les offres des services afin de sauver les vies des personnes le plutôt possible.

2. Modélisation du système

2.1. Diagramme de cas d'utilisateur :

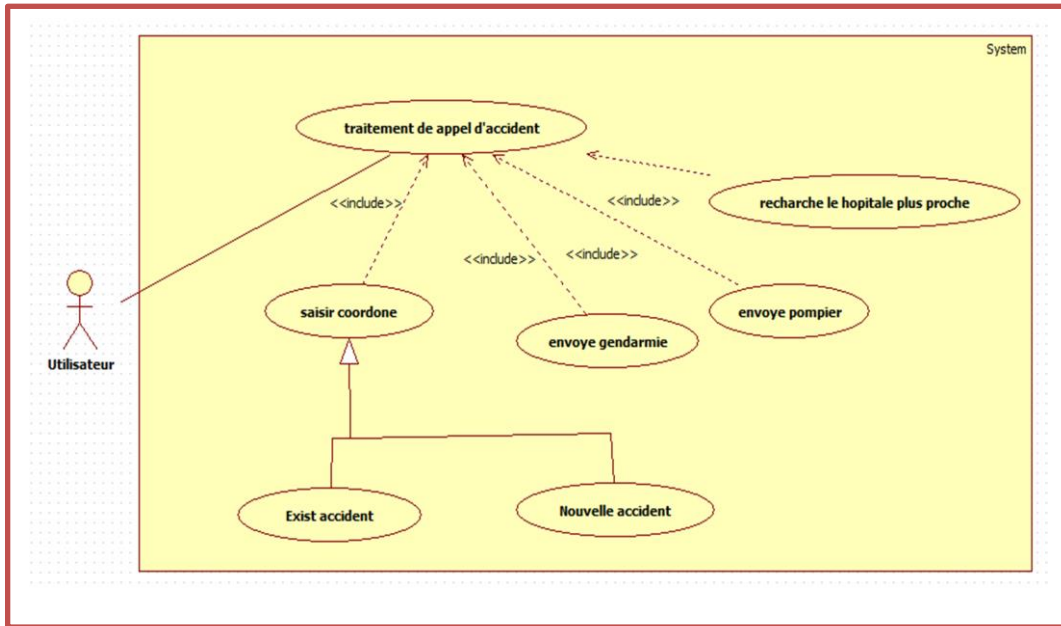


Figure II-1: diagramme de cas d'utilisation .

2.2. Diagramme de séquence :

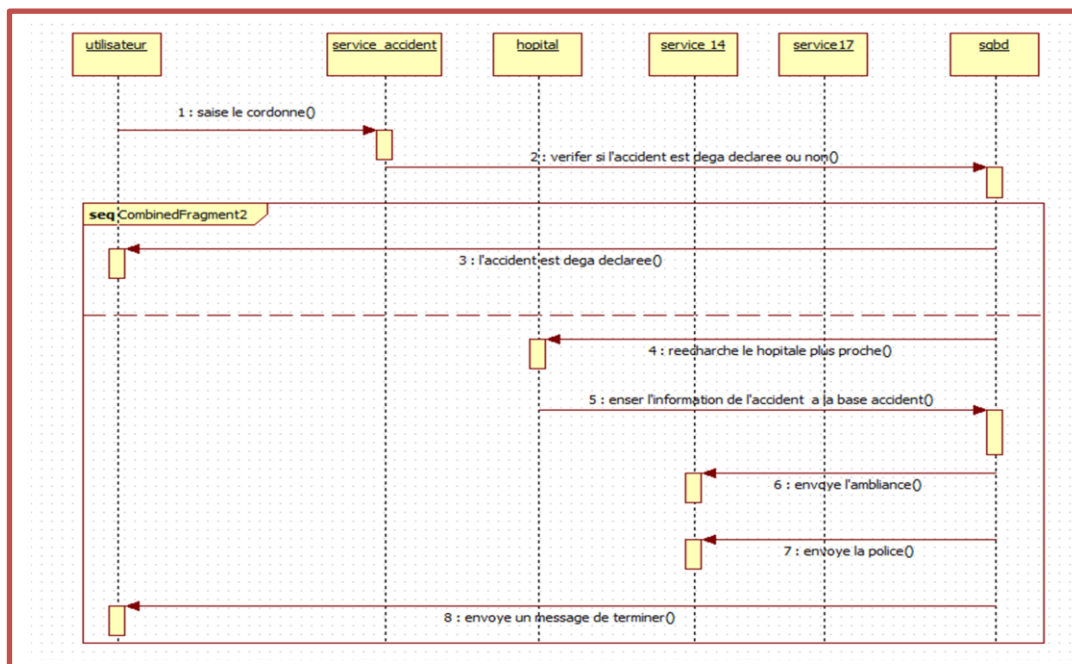


Figure II-2 : Diagramme de séquence .

2.3. Diagramme d'activité du processus :

Dans la Figure (II-2) suivante le diagramme d'activité montre l'enchaînement des activités qui est illustré :

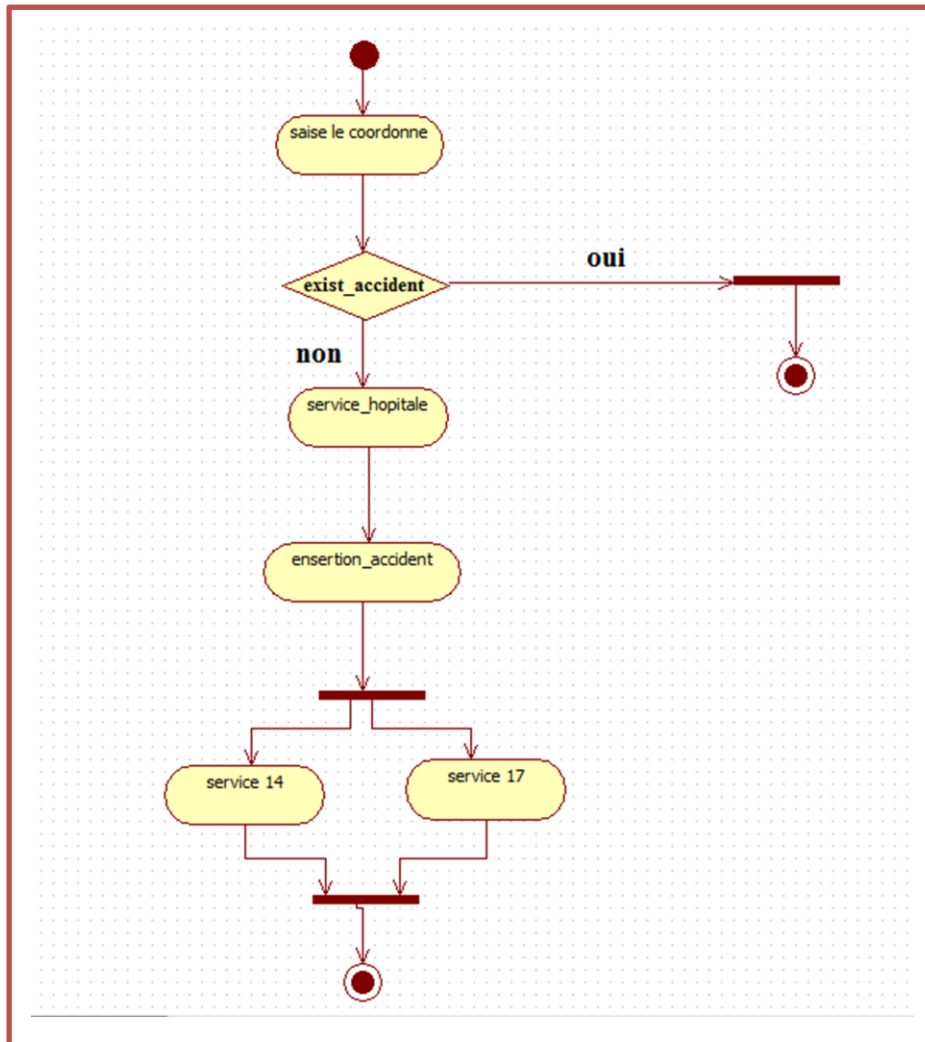


Figure II-3 : Diagramme de activité de processus.

3. Fonctionnement du système d'appel d'accident :

- 1- avant signaler un accident, , collecte les informations sur l'accident si l'adresse de l'accident n'est pas en coordonné GPS, alors il va la traduire en adresse GPS.
- 2- Le utilisateur saisie les cordonnée GPS (x,y) avec la date de l'accident .
- 3- On doit vérifier si cet accident est déjà pris en compte ou non.
- 4- Si c'est une nouvelle déclaration, alors, on va chercher l'hôpital le plus proche du lieu de l'accident.

5- insérer l'accident dans la base des accidents.

Ensuite on va informer simultanément

6- le service Pompier pour envoyer une Ambulance .

7- le service Gendarmerie pour envoyer la patrouille qui va faire l'enquête.

8- enfin on va informer un message de confirmation de la prise en compte de l'accident .

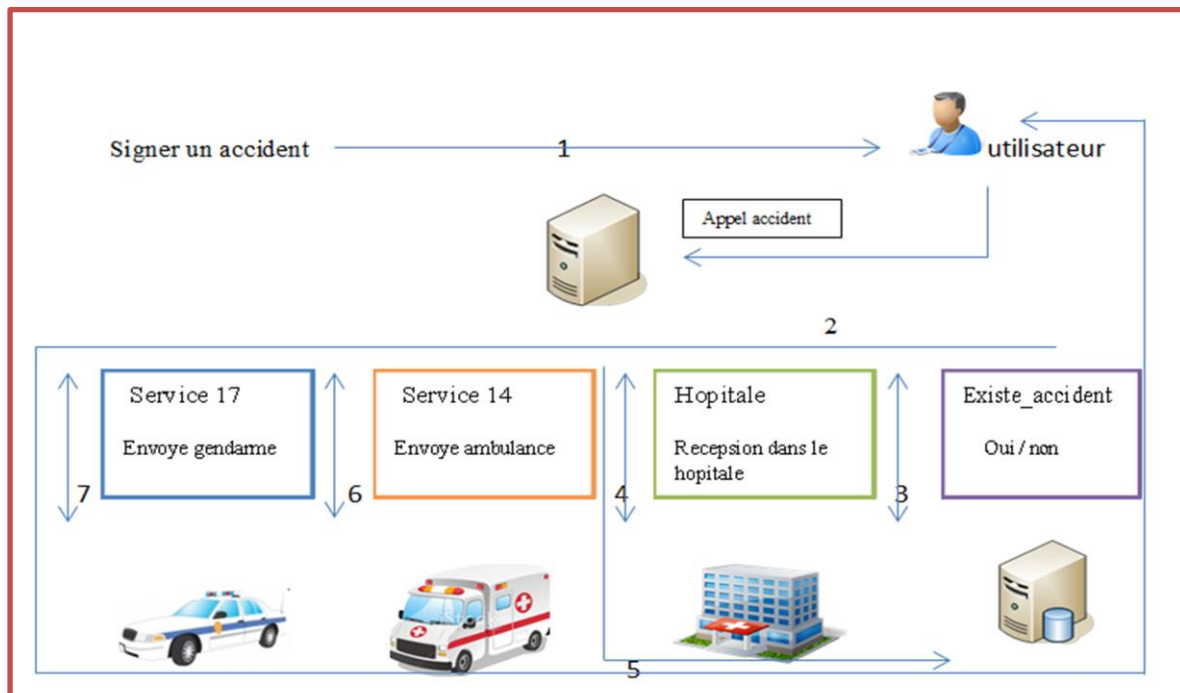


Figure II-4 : fonctionnement de système .

4. Définition des Services Web du Système :

Le service web composite de system est quatre services web qui sont .

- Service accident
- Service Pompier_ Hôpital
- Service 17

4.1. Service accident :

Ce system offre deux opérations :

Exist_accident : Permet de vérifier si l'accident est déjà declaree.

Exist_accident (GPS_X ,GPS_Y, date_accident) → boolean .

Ensertion_accident : permet de ensertion de l'accident dans la base des accident .

Ensertion_accident(GPS_Xacc ,GPS_Yacc, date_accident ,Nom_Hopitale) → String .

Ce service contient une Base de données : base_accident

Champs	Type
Id	Int
GPS_x	Int
GPS_y	Int
Date_accident	String
Nom_hopitale	String

Tableaux II-1 : tableaux de accident dans la base de donnée .

4.2. Service Pompier_ Hôpital :

Ce system offre deux service

Service ser_hopitale :permet de calcule le hopitale plus proche a l'accident .

Hopitale (GPS_X ,GPS_Y, date_accident) → String .

Ce service contient une Base de données : hopitale

Champs	Type
Nom_hopitale	String
GPS_hopitaleX	Int
GPS_hopitaleY	Int

Tableaux II-2 : tableaux de hôpital dans la base de donnée .

Service14 : permet d'informer les pompiers pour envoyer une ambulance.

Pompier (GPS_X ,GPS_Y, date_accident) → String .

4.3. Service 17 :

Service17 : permet d'envoyé le informationna de l'accident à la police .

Ser17 (GPS_X ,GPS_Y, date_accident , nom_hopitale) → String

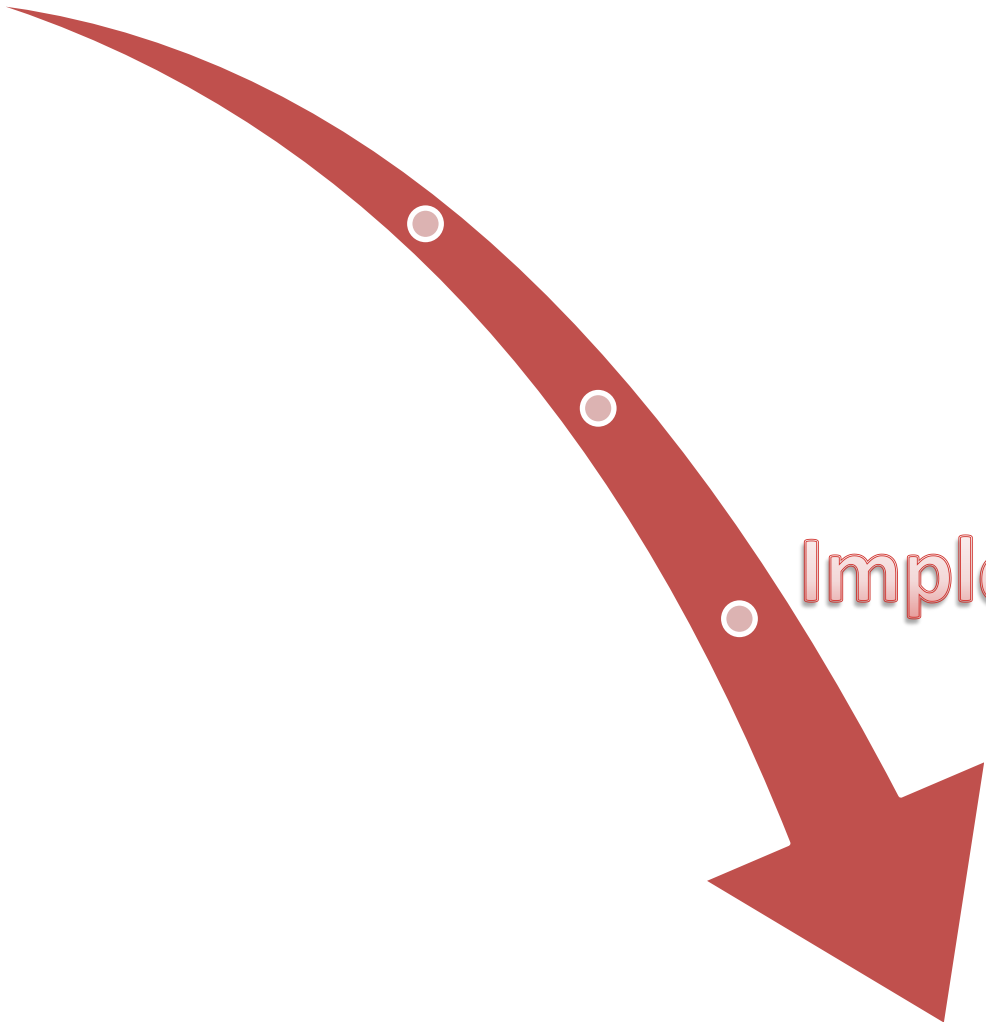
Conclusion

La composition de services web a pour but d'utiliser les compétences de plusieurs services afin de résoudre un problème qu'aucun ne saurait résoudre individuellement.

Le résultat de cette composition est un enchaînement de services web qui permet de définir la façon dont les données calculées par les uns sont consommées par les autres.

Nous avons présenté dans cette partie la conception de notre Système qui consiste en une composition statique de plusieurs services web qui permet de développer un service web composite.

Nous avons présenté également, les différentes fonctions proposées de notre système. Dans le chapitre qui suit, nous allons parler de différents outils nécessaires pour la mise en œuvre de cette conception ensuite nous allons montrer l'implémentation et le fonctionnement de notre système par un exemple.



Implementation

Introduction

Nous avons défini dans le parti précédent les principaux composants de notre système de traitement Appel accident, comme nous l'avons expliqué, il faut développer quatre services Web à savoir : Service Base d'accident, Service Hôpital, Service 14 et Service 17. Le service Appel accident est le résultat de la composition de ces quatre services en utilisant le principe d'orchestration.

Nous présentons dans ce parti la mise en œuvre de ce système, en faisant une brève présentation des outils de développement utilisés, puis la réalisation de notre processus métier.

Nous terminerons par la présentation de quelque extrait de code et un petit exemple d'exécution du système.

1. Les Outils utilisés

Le Langage de programmation

Pour l'implémentation de notre système nous avons opté pour le langage Java (version du JDK : 1.6). Ce langage est apparu vers la fin de 1995 et obtint rapidement un énorme succès. Nous nous sommes donc orientés vers ce langage pour le développement de notre système qui présente beaucoup d'avantages [19]:

- ✚ Java assure une totale indépendance des applications vis-à-vis de l'environnement d'exécution : c'est à dire que toute machine supportant Java est en mesure d'exécuter un programme sans aucune adaptation (ni recompilation, ni paramétrage de variables d'environnement);
- ✚ Java est un langage orienté objet, c'est à dire que nous n'allons pas manipuler des fonctions et des procédures mais des objets qui vont s'échanger des messages. Le principal avantage est que l'on peut réaliser une programmation modulaire : tous les objets peuvent être mis au point séparément ;

Java nous permet un accès simplifié aux bases de données à travers un pilote spécifique au SGBD ;

En ce qui concerne l'environnement de travail, nous avons choisi NETBEANS (version 6.7.1 de constructeur Sun.) qui est un environnement de développement intégré (EDI).

Description de NetBeans IDE 6.7

NetBeans est un environnement de développement intégré (IDE) open source. Il est développé par Sun et se trouve sous licence CDDL (Common Development and Distribution License). En plus de Java, il propose tous les outils nécessaires à la création d'applications professionnelles pour les particuliers, les entreprises, le web et les applications mobiles avec le langage C / C + +, et même les langages dynamiques tels que PHP, Javascript, Groovy, Ruby XML et HTML. NetBeans IDE est facile à installer et à utiliser. Il comprend toutes les caractéristiques d'un IDE moderne (coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web, etc.) [18].

La version 6.7.1 de NetBeans est téléchargeable à :

<http://www.netbeans.org/kb/60/soa/casa-bc-bc.html>

NetBeans possède de nombreux points forts qui sont à l'origine de son énorme succès dont les principaux sont [18] :

Un environnement de développement intégré comme beaucoup d'autres plateformes (Eclipse, JBuilder.)

Caractérisé par un aspect modulaire (constitué d'un ensemble de modules indépendants les uns des autres appelés plug-ins). Ce concept permet de favoriser la création des extensions de la plateforme et de permettre à des tierces parties de développer de nouvelles fonctionnalités qui ne sont pas fournies en standard. Autrement dit, offre aux développeurs la possibilité de personnaliser leurs environnements.

Supporte tout le cycle de vie du développement du fait qu'il intègre des dispositifs pour la modélisation, la programmation, le débogage, le test et le déploiement des applications.

Serveur SGBD Mysql :

MySQL est un gestionnaire de base de données libre. Il est très utilisé dans les projets libres et dans le milieu industriel. MySQL est un SGBD relationnelle développé dans un souci de performances élevées. Il est multiutilisateurs [4].

MySQL offre l'avantage d'être portable, c'est-à-dire, il est compilé sur plusieurs plateformes comme Windows, Linux, Mac OS, Solaris,...etc. Son architecture rend son utilisation extrêmement facile, rapide et robuste [4].

La base de données MySQL est devenue la base de données open source la plus populaire au monde grâce à sa haute performance, sa fiabilité et sa simplicité d'utilisation.

La version Mysql-5.1.38-win32 est téléchargeable à :

<http://www.commentcamarche.net/download/telecharger-84-mysql>.

Pour connecter à Mysql serveur, on utilise Mysql Connector.

. MySQL connector :

Pour connecter avec MySQL, on a utilisé le pilote (driver) MySQL connector version mysql-connector-java-5.1.15.

Pour intégrer Mysql connector dans l'environnement NetBeans IDE il suffit d'ajouter le fichier jar nécessaire voir Figure V-2:

Serveur d'application Glass Fish :

GlassFish est le nom du serveur d'applications Open Source Java EE 5 et qui sert de fondation au produit Sun Java System Application Server de Sun Microsystems. Sa partie Toplink devient d'Oracle. C'est la réponse aux développeurs Java désireux d'accéder aux sources et de contribuer au développement des serveurs d'applications de nouvelle génération de Sun [20].

GlassFish est sous double licence CDDL et GPLv2 et il est certifié Java EE 5 (source GlassFish) [21].

OpenESB (via Glassfish 2.1) et Netbeans 6.7.1 est téléchargeable à : <https://glassfish.dev.java.net>.

2. Déploiement de service composite

2.1. Test le web service :

Test web service accident :



The screenshot shows a web browser window with the URL `localhost:8080/Accident/ServiceAccidentService?Tester`. The page title is **ServiceAccidentService Web Service Tester**. Below the title, there is a description: "This form will allow you to test your web service implementation (WSDL File)". A note states: "To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name." Under the heading "Methods :", two methods are listed:

```
public abstract java.lang.Boolean sa.ServiceAccident.existAccident(int,int,java.lang.String)
existAccident ( )

public abstract java.lang.String sa.ServiceAccident.insertionAccident(int,int,java.lang.String,java.lang.String)
insertionAccident ( )
```

Figure III-1: test le service « accident » .

Test web service “service14” :



The screenshot shows a web browser window with the URL `localhost:8080/pompier_hopitale/service14Service?Tester`. The page title is **service14Service Web Service Tester**. Below the title, there is a description: "This form will allow you to test your web service implementation (WSDL File)". A note states: "To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name." Under the heading "Methods :", one method is listed:

```
public abstract java.lang.String s14.Service14.pompier(int,int,java.lang.String,java.lang.String)
pompier ( )
```

Figure III-2: test le service « service 14 ».

Test web service “ser_hopitale” :

localhost:8080/pompier_hopitale/ser_hopitaleService?Tester

ser_hopitaleService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

```
public abstract java.lang.String sh.SerHopitale.hopitale(int,int,java.lang.String)
```

hopitale ()

Figure III-3: test le service « ser_hopitalee ».

Test web service “service17” :

localhost:8080/service17/ser02?Tester

ser02 Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

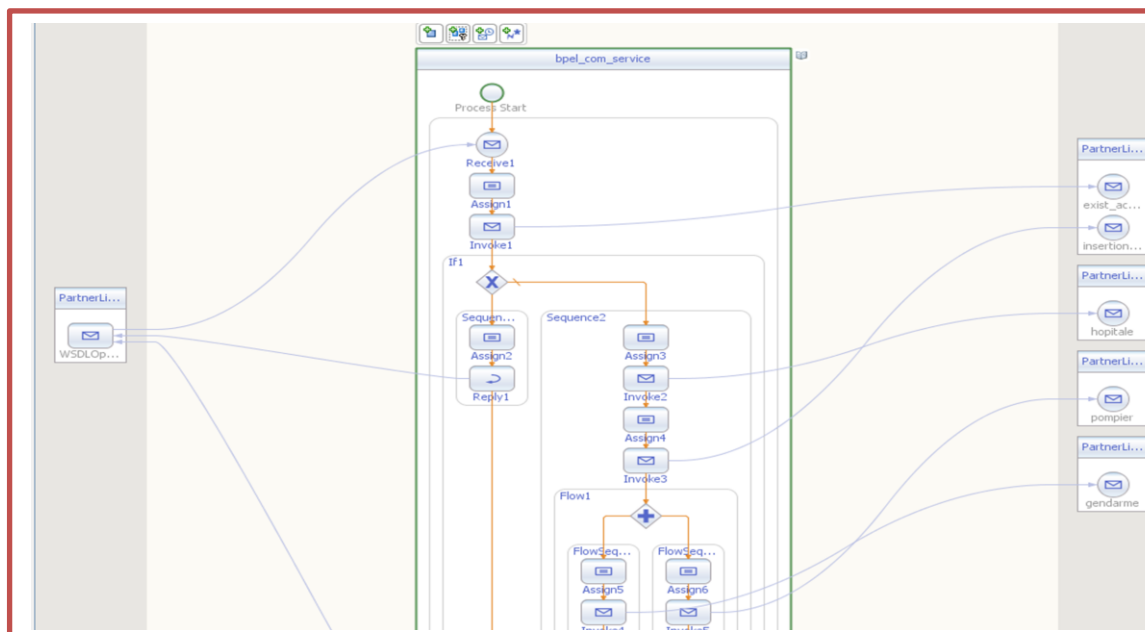
Methods :

```
public abstract java.lang.String s02.Ser02.gendarme(int,int,java.lang.String)
```

gendarme ()

Figure III-4: test le service « service 17 » .

2.2. Design de BPEL :



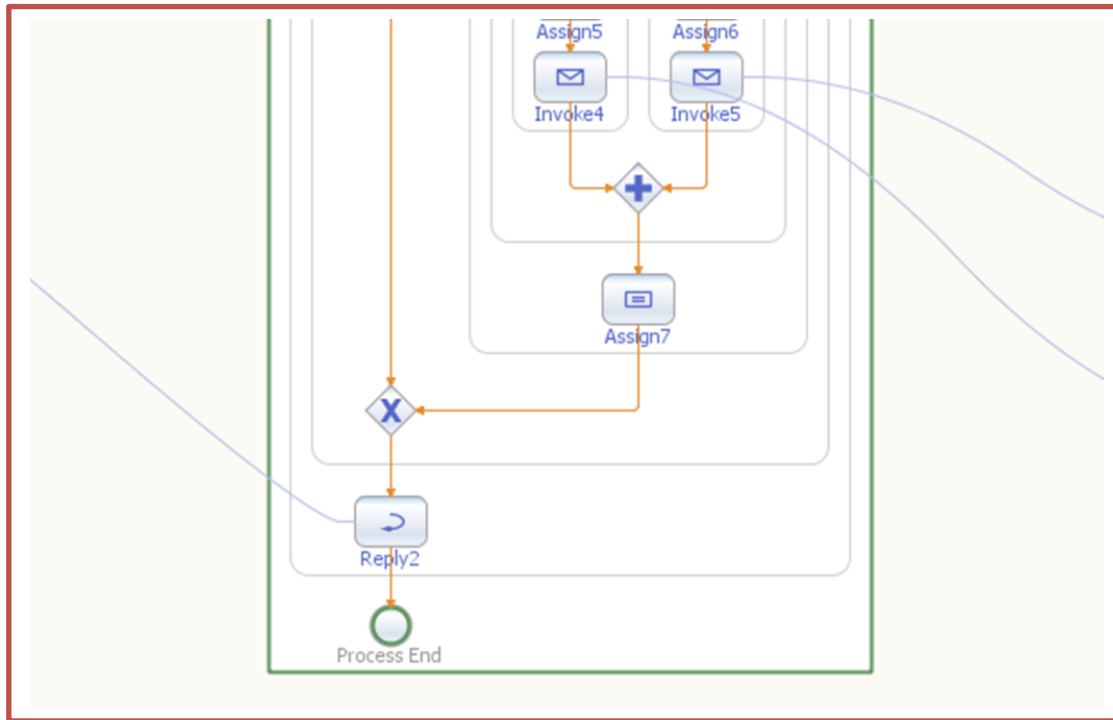


Figure III-5: PBEL Module .

2.3. Composition de service :

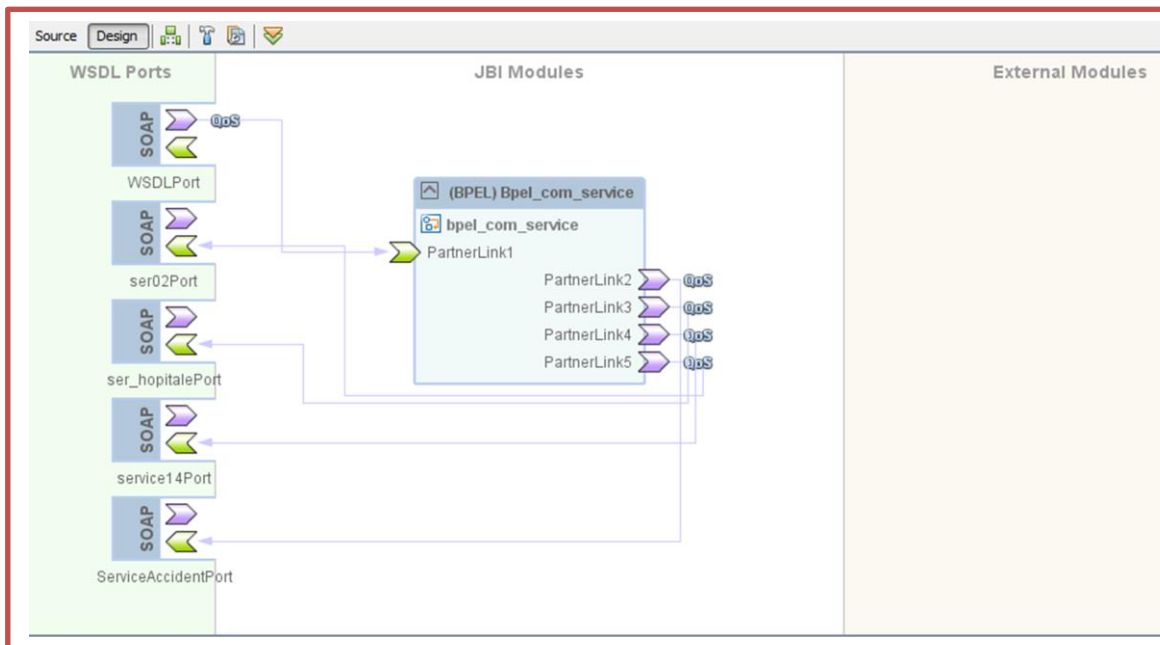


Figure III-6: composite application .

2.4. Test de l'application de composition (Composite Application)

```
<soapenv:Envelope xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wSDL="http://j2ee.netbeans.org/wsdl/Bpel_com_service/WSDL">
  <soapenv:Body>
    <wSDL:WSDLOperation>
      <GPS_X>20</GPS_X>
      <GPS_Y>20</GPS_Y>
      <part1>20-01-2014</part1>
    </wSDL:WSDLOperation>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure III-7: test le composite application « request » .

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <m:WSDLOperationResponse xmlns:m="http://j2ee.netbeans.org/wsdl/Bpel_com_service/WSDL">
      <part1>ok envoie le gendarme envoie l'ambulance</part1>
    </m:WSDLOperationResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure III-8: test le composite application «réponde » .

2.5. Exécution de L'application



Figure III-9: interface de le utilisateur



Figure III-10: interface de saisir le cordonnée

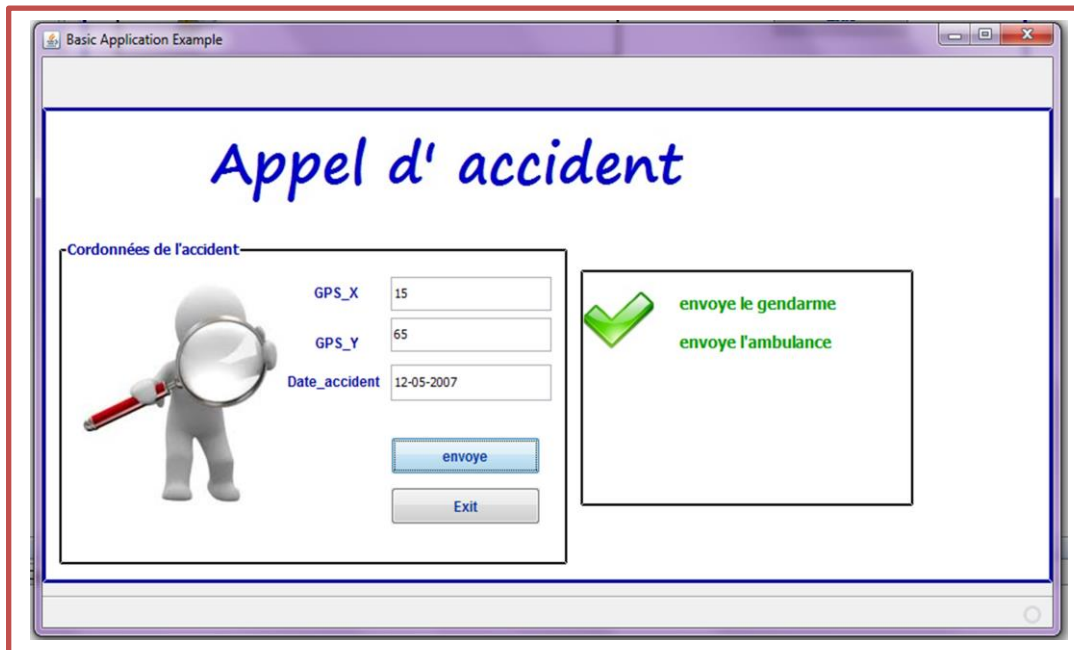


Figure III-11: résultat si l'accident est en nouvelle déclaration



Figure III-12: résultat si l'accident est déjà déclarée

Conclusion

Nous avons présenté dans ce parti les principaux outils que nous avons utilisé dans notre démarche : l'environnement de développement intègre Netbeans version 7.6.1 (qui fournit l'éditeur graphique du processus Bpel, l'environnement de langage java, outils pour la création des service web...), le serveur GlasseFish 2.1 qui intègre le moteur open ESB et le serveur SGBD Mysql la version utilisée est Mysql-5.1.38-win32 et pour connecter avec MySQL, on a utilisé le pilote (driver) MySQL connector version mysql-connector-java-5.1.15.

Puis nous sommes passés à la présentation de notre processus métier Service Appel accident a travers les différentes parties du BPEL comme les attributs supérieurs, les liens de partenaire, les activités et les variables, on détaillons un peu chaque partie avec un extrait de code du fichier BPEL et fichier WSDL et des captures d'écran d'utilisation des outils d'aide à la construction du contenu des différentes activités.

Et à la dernière partie nous avons expliqué le fonctionnement de notre interface graphique. Depuis où le réceptionniste fait la saisie des coordonnées jusqu'à l'affichage des différentes réponses.

Conclusion générale

Ce projet de fin d'étude nous a permis d'aborder le domaine des processus métier et celui des services web et d'apprendre comment passer d'un modèle de processus conceptuel vers une implémentation technique. C'est ce qu'on appelle une gestion de cycle de vie d'un processus métier.

Notre objectif étant la réalisation d'un processus métier en utilisant la technique de l'orchestration des services web avec une composition statique et automatique en utilisant le langage BPEL pour un système de traitement d'appel d'accident .

Afin d'atteindre ce but, nous avons commencé par étudier la nouvelle technologie des Services web et les technologies qui permettent leurs mises en œuvre. Ensuite nous avons fait une étude approfondie des types et techniques disponibles pour la composition statique de services web.

Nous avons fait une conception qui consiste sur une approche de composition processus métier de services de manière automatique et statique car les services participants sont connus à l'avance.

Nous avons également été confrontées à étudier un nouveau langage à savoir BPEL pour le développement de notre processus métier.

Cependant, notre travail peut être amélioré de plusieurs manières. Nous pouvons citer par exemple les perspectives suivantes :

- ✚ Le programme peut être converti de la plaine à la page Web du programme.
- ✚ Vous pouvez ajouter des modifications à la base de données se développent pour inclure plus de régions.
- ✚ Le programme peut être généralisé pour appliquer aux téléphones pour vitesse en réponse à l'appel.

Bibliographie

- [1] D.Booth, H. Haas, F. McCabe, E. Newcomer, M.Champion, C.Ferris, et D. Orchard, Web Service Architecture, W3C Working Group Note, Février 2004.
- [2] <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>, (dernière visite 09/05/2013).
Uniform Resource Identifier, édité le 03/05/2013,
http://fr.wikipedia.org/wiki/Uniform_Resource_Identifier, (dernière visite 09/05/2013).
- [3] M.Hamaz ,N. Ikhlef , Thème : « Composition personnalisée des services web dans le domaine de e_learning ». Mémoire de fin d'études ingénieur, (INI) , juin 2009.
- [4] I.Khlfaoui , K.Haliche, Thème : « Conception et réalisation d'un système pour la Composition de services web » , *Mémoire de fin d'études ingénieur* , (INI) , juin 2008.
- [5] <http://www.gchagnon.fr/cours/xml/serviceweb.html>
- [6] F.Benseddik, N.Ramdani, Thème : « Un système pour la composition automatique des services web basée sur leur comportement », Mémoire de fin d'études ingénieur, (ESI), juin 2009.
- [7] D. Booth, H. Haas, F. McCabe, and E. Newcomer. "Web services architecture", aout 2003. <http://www.w3.org/TR/2003/WD-ws-arch-20030808/>.
- [8] Douglas K. Barry. "Web Services and Service-Oriented Architectures : The Savvy Manager's Guide". Morgan Kaufmann, 2003.
- [9] K. Heather. "Web services conceptual architecture(w sca 1.0) ", may 2001. <http://www-306.ibm.com/software/solutions/webservices/pdf/WSCA.pdf>.
- [10] Humberto, Cervantes - 2004 - "Vers un modèle à composants orienté services pour supporter la disponibilité dynamique", Université Joseph Fourier - Grenoble 1.
- [11] Mark Endrei, Jenny Ang, Ali Arsanjani, Sook Chua, Philippe Comte, Pål Krogdahl, Min Luo, Tony Newling - 2004 – "Pattern: Service Oriented Architecture and Web Services", IBM INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION , ISBN 073845317X
- [12] <http://www.w3.org/2002/ws/>
- [13] P. Kellert and F. Toumani. "Les web services sémantiques". In Web sémantique, Action spéci_que 32 CNRS/STIC, October 2003.
- [14] Sami Bhiri; Thèse Doctorat "Approche Transactionnelle pour Assurer des Compositions Fiables de Services Web " université Henri Poincaré- Nancy 1. Octobre 2005.

- [15] H.FOSTER, S.UCHITEL, J.MAGEE, J.KRAMER. Model-based Verification of Web Service Composition, Oct. 2003.
- [16] M. JURIC, BPEL and JAVA. April 2005
- [17] C.PELTZ. Web Service Orchestration and Choreography – A look at WSCI and BPEL4WS. July 2003.
- [18] <http://www.netbeans.org>
- [19] <http://www.java.com>
- [20] <https://glassfish.dev.java.net>.