REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre :GLSD 10 /M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Génie logiciel et Systèmes Distribués(GLSD)**

Titre : Génération de cas de test pour les services Web en utilisant la perturbation des données

---

Par :

SOUDADI BAHIA

Soutenu le :01/07/2021 devant le jury composé de :

Nom Prénom	grade	Président
MerabetRabiya	grade	Rapporteur
Nom Prénom	grade	Examinateur

Année universitaire 2020-2021

Table des matières

Listedesfigures :

Liste des tableaux :

List des abrévations :

ADO	ActiveX Data Objects
API	Application programming interface
ASP	Active Server Page
ASP	ActiveServerPage
BPEL	Business Process ExecutionLanguage
DCP	Perturbations de communication de données
DTD	Document Type Definition
EDI	Electronic Data Interchange
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated DevelopmentEnvironment
IEEE	Institute of Electrical and Electronics Engineer
IIS	Internet Information Server
JSON	JavaScript Object Notation
MRCS	Robot Mars Système de communication
OWL-S	Ontology Web Language for Services
PVD	Perturbation de la valeur des données
REST	Representational State Transfer
RPC	RemoteProcedure Calls
RTG	grammaire d'arbre régulier
SGBDR	gestiondebasesdedonnéesrelationnelles
SOA	architecture orientée services
SOAP	SimpleobjectAccessProtocol
SQL	StructuredQueryLanguage
SQL	StructuredQueryLanguage
SUT	Service sous test
UDD	Universal Description, Discovery and Integration
UML	UnifiedModellingLanguage
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
W3C	World WideWeb
WSDL	WebServicesDescriptionLanguage
WS-I	Web Services Interoperability
WS-TAXI	A WSDL-based Testing Tool for Web Services
XML	Extensible Markup Language
XSD	XML SchemaDefinition

REMERCIEMENTS

Nous remercions tout d'abord Dieu qui nous a donné la foi et le courage pour accomplir ce projet.

On tient aussi à exprimer nos vifs remerciements à notre encadreur ***MERABET RABIYA*** *qui nous a aidés à réaliser ce travail.*

Nos profondes gratitudes vont également à l'attention des membres de jury qui nous ferons honneur d'examiner notre travail, ainsi tous les enseignants qui ont contribués à notre formation.

A fin de ne pas oublier personne, nous remercions tous ceux qui ont contribués de près ou de loin à l'élaboration de cette mémoire.

SOUDADI BAHIA

DEDICACE

Dédicace A la première femme que mon regard a pu voir, toi ma très chère mère qui a sacrifié sa vie pour l'éducation de ses enfants, quoi que je fasse je pourrai jamais t'offrir ce que tu ma procuré. Laisse moi te dédier ce modeste travail et que dieu vous garde pour nous.

A toute ma famille A tous mes proches grands et petits.

A tous ceux qui m'ont aidé de prés ou de loin dans mes études.

Résumé

Les services Web ont le potentiel de réduire considérablement la complexité et les coûts des projets d'intégration de logiciels.

La différence la plus évidente et peut-être la plus significative entre Les services Web et les applications traditionnelles que les services web utilisent une infrastructure de communication commune, XML et SOAP, pour communiquer via Internet.

La méthode de la communication introduit des complexités aux problèmes de vérifier et valider des services Web qui n'existent pas dans les logiciels traditionnels.

Ce mémoire présente l'approche de teste les services web en fonction de la perturbation des données avec architecture soap qui permet de manipuler un compte bancaire avec des déférentes opérations qui en peut le faire, Les tests sont manipulés manuellement.

Les messages XML sont modifiés en fonction des règles définies sur les grammaires des messages, puis utilisées comme tests.

La perturbation des données utilise deux méthodes pour tester les services Web: Perturbation valeur de données et perturbation d'interaction.

La perturbation par valeur des données modifie les valeurs en fonction du type de données.

La perturbation d'interaction classe les messages de communication en deux catégories: communication RPC et communication de données.

mots-clés: Web Services, XML, SOAP, Perturbation de données.

ABSTRACT

Web services have the potential to dramatically reduce the complexities and costs of software integration projects.

The most obvious and perhaps most significant difference between Web services and traditional applications is that Web services use a common communication infrastructure, XML and SOAP, to communicate through the Internet.

The method of communication introduces complexities to the problems of verifying and validating Web services that do not exist in traditional software.

This thesis presents the approach of testing web services according to the disruption of data with soap architecture that allows you to manipulate a bank account with various operations that can do so. The tests are handled manually.Existing XML messages are modified based on rules defined on the message grammars, and then used as tests.

Data perturbation uses two methods to test Web services: data value perturbation and interaction perturbation. Data value perturbation modifies values according to the data type.

Interaction perturbation classifies the communication messages into two categories: RPC communication and data communication.

keywords: Web Services, XML, SOAP, Data perturbation

1. Introduction :

Ce chapitre a pour objectif de présenter le cadre de départ de nos travaux. Dans un premier temps, nous introduisons les concepts de base. Nous définissons ce que nous entendons par architecture orientée service, et décrivons les caractéristiques qui les rendent attirantes pour les concepteurs d'architectures distribuées.

1.1- Présentationducontexte :

Les services Web ne sont pas encore largement utilisés en raison de problèmes de sécurité. Mais il y a un barrage routier encore plus grand attendre juste en bas de la route - ça s'appelle la confiance. Le gros problème est le service fonctionnera-t-il correctement chaque le moment où j'en ai besoin? Pour l'instant, peu d'entre eux réfléchissent aux problèmes des tests et de la certification. Nous suggestions que les tests et la certification des services Web ne se déroulent pas comme d'habitude et que de nouvelles solutions sont nécessaires pour donnent l'assurance que les services sont vraiment fiables.

L'architecture orientée services permet d'être une solution très efficace pour ces systèmes, enaméliorant leur qualité et en simplifiant leur intégration dans l'infrastructure informatique del'entreprise, via l'utilisation de composants réutilisables et distribués. Ces composants, baséssurl'infrastructured'internet,sont appeléservicesWeb.

Un service Web est une technologie qui permet aux entreprises de faire communiquer leurssystèmesd'informationsavecceuxdeleursclientsoupartenaires,viainternet.Cettetechnologie baséesurlestandardXMLesttotalementindépendantedumatériel,oudu langage de programmation utilisé ce qui permet de facilement la mettre en œuvre.

Pour garantir la fiabilité, ces processus de qualité intègrentdansleurméthodologie un ensemble d'activitésdetests.

En effet, les services Web n'appartiennent pas directement à l'application qui les invoque. Ilssont accessibles à travers plusieurs couches (SOAP, HTTP/HTTPS, couches client, processeurSOAP), ce qui réduit leur visibilité.

Le Web a évolué très rapidement et la technologie Internet a été adoptée par des utilisateurs appartenant à des domaines de plus en plus diversifiés. Comparées aux applications

traditionnelles, les applications Web présentent plusieurs caractéristiques propres. Il existe diverses formes au test web: le test fonctionnel à travers le test de robustese.

Les méthodes qui visent à tester les aspects non fonctionnels des services Web, tels que la sécurité ou la qualité de service, sont extérieuresa portée de cette mémoire.

Dans cette mémoire, la catégorisation de la recherche dans les tests de services Web basés sur les pannes est effectuée en fonction du niveau auquel les défauts sont générés.
Les tests basés sur les pannes des services Web peuvent être regroupés en
Trois catégories différentes selon le niveau auquel les défauts sont appliqués :

- Perturbations des messages XML / SOAP.
- injection de défauts au niveau du réseau.
- mutation des spécifications du service Web.

Cettemémoireadonc pourbut:

• d'étudierprincipalement letestdeservicesWeb, par perturbation de données et leur technique (DVP ; RCP ; DCP), Un des premiers exemples de perturbation SOAP qui proposé par Offutt et Xu en 2004.

• d'expliquer laméthodesdetest par implémentation un système à base de service avec une application qui génère manuellement la technique de perturbation de donnée avec des jeux de test quiprennentencomptel'environnementSOAPdes servicesWeb pour lesvalider au mieux.

1.2- Cycle de vie d'un logiciel :

Le « **cycle de vie d'un logiciel** » (en anglais *software lifecycle*), désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. L'objectif d'un tel découpage est de permettre de définir des repères intermédiaires permettant la **validation** du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la **vérification** du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre.

L'origine de ce découpage provient du constat que les erreurs ont un coût d'autant plus élevé qu'elles sont détectées tardivement dans le processus de réalisation. Le cycle de vie permet de détecter les erreurs au plus tôt et ainsi de maîtriser la qualité du logiciel, les délais de sa réalisation et les coûts associés.

Le cycle de vie du logiciel comprend généralement a minima les activités suivantes :

- **Définition des objectifs**, consistant à définir la finalité du projet et son inscription dans une stratégie globale.
- **Analyse des besoins et faisabilité**, c'est-à-dire l'expression, le recueil et la formalisation des besoins du demandeur (le client) et de l'ensemble des contraintes.
- **Conception générale**. Il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel.
- **Conception détaillée**, consistant à définir précisément chaque sous-ensemble du logiciel.
- **Codage** (Implémentation ou programmation), soit la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception.
- **Tests unitaires**, permettant de vérifier individuellement que chaque sous- ensemble du logiciel est implémentée conformément aux spécifications. Cestestspeuventêtredivers:liésauxperformances,àla robustesse,ainsiqu'àlaconformitédel'implantationparrapportàsaspécification.C'estdans un cadrede ces typesde testques'inscrit cettemémoire.

- **Intégration**, dont l'objectif est de s'assurer de l'interfaçage des différents éléments (modules) du logiciel. Elle fait l'objet de *tests d'intégration* consignés dans un document.
- **Qualification** (ou *recette*), c'est-à-dire la vérification de la conformité du logiciel aux spécifications initiales.
- **Documentation**, visant à produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs.
- **Mise en production**,
- **Maintenance**, comprenant toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.**[2]**

La séquence et la présence de chacune de ces activités dans le cycle de vie dépend du choix d'un modèle de cycle de vie entre le client et l'équipe de développement. Il définit des phases séquentielles à l'issue de chacune desquelles des documents sont produits pour en vérifier la conformité.

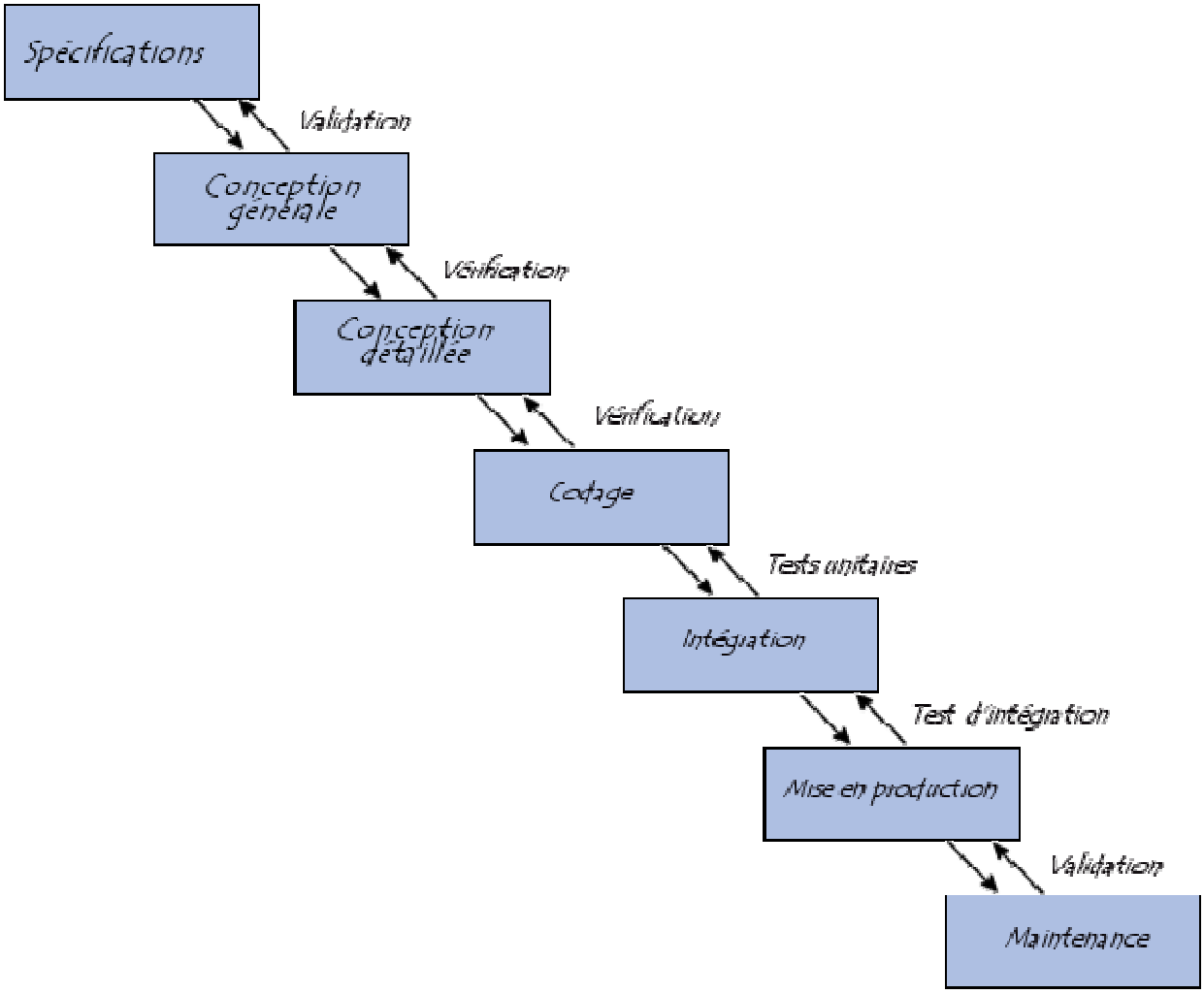


Figure 1-1: Cycle de vie du logiciel [2]

1.3- Plan delamémoire :

Chapitre 01 — Introduction générale :

Ce chapitre présente un contexte général sur le mémoire et leur but, ainsi le plan du mémoire,et liste des travaux liés à cette mémoire.

Chapitre2—Service web & les méthodologies de test :

Ce chapitre présente l'historique, les concepts et techniques des services web ainsi que lesmécanismes de composition de services,Le chapitre 2 est consacré en premier lieu à la définition de modèles de description formelle.Ildétailleensuitelesarchitecturesdetestsetlesoutils de test les plus connu.

Chapitre3—Techniques de perturbation des données:

Ce chapitre propose un état de l'art détaillé sur lestechniques de perturbation des données, il expose quelques méthodes de test appliquées dans le domaine de services Weben prenant en compte l'environnement SOAP.

Chapitre4—: conception & implémentation

Le chapitre 4 présente la conception de système avec des différents diagrammes et classe utilise dans l'implémentation avec des exemples de test et les résultats capturée qui expliquent les déférents cas de test généré.

Résultat Empiriques

Conclusion général et travaux futur

Références bibliographique

1.4- Liste des travaux liés à cette mémoire :

Nous présentons dans cette partie un ensemble d'auteurs ont abordé le test par perturbation

1.4.1 Xu et coll :

`Xu et coll` : proposent une approche où la perturbation est appliquée aux schémas XML afin de générer les cas de test. Xu et coll. définir des opérateurs de perturbation de schéma XML pour créer des messages XML non valides en insérer ou supprimer des champs de données. [14]

1.4.2 Almedia et Vergilio :

`Almedia et Vergilio` : adoptent également la même approche et proposent un outil appelé SMAT-WS qui automatise le processus de test. Almedia et Vergilio présentent également quelques nouveaux opérateurs de perturbation pour la perturbation XML. [15]

1.4.3 Samer et Munro :

`Samer et Munro` : Testent la robustesse des services en violer les spécifications des paramètres d'entrée des fichiers WSDL. L'approche de Samer et Munro peut tester les deux services Web lui-même et la plate-forme sur laquelle réside le service**. [16]**

1.4.4 Zhang et Zhang :

`Zhang et Zhang` : proposent une limite valoriser les tests d'injection de pannes afin de vous aider à sélectionner les services Web fiables. De même, [17]

1.4.5 Vieira et Al :

Vieira et Al : proposent un cadre qui effectue l'injection de fautes dans les messages SOAP capturés. [18]

1.4.6 Martin et coll :

Martin et coll : proposent un cadre appelé WebSob qui teste la robustesse des services Web. WebSob teste le service Web méthodes avec des paramètres extrêmes ou spéciaux. [19]

1.4.7 Tsai et coll :

Tsai et coll : aborder les problèmes du classement des données de test et des tests basés sur les pannes dans une approche unique.

proposent une approche basée sur l'analyse des expressions booléennes qui peut générer à la fois vrai et faux cas de test. L'approche proposée est soutenue par un cadre qui peut classer les cas de test selon la probabilité des cas de test de révéler des erreurs. [20]

Les résultats :

Les résultats des perturbations SOAP prouvent l'efficacité de cette approche pour révéler les failles dans le web prestations de service. Par exemple, pendant les expériences Offutt et Xu., 18 défauts sont insérés dans le robot Mars Système de communication (MRCS) et 100 tests DVP, 15 RCP et 27 DCP sont générés.

Les tests générés atteignent un taux de détection des défauts de 78% dans les défauts prédéfinis (14 sur 18) et a également révélé deux défauts naturels. Xu et coll.

également expérimenté sur MRCS et en plus sur l'application de gestion de la chaîne d'approvisionnement de WS-I et a atteint 33% de détection de défaut.

Almedia et Vergilio ont mené leurs expériences sur un système composé de neufservices Web et a révélé 49 défauts dont 18 sont générés par SMAT-WS. Vieira et coll. Expérimenté sur 21 services Web publics et observé un grand nombre de pannes; cependant, 7 de ces services ont montré pas de problèmes de robustesse. Vieira et coll. soulignent également qu'un nombre significatif d'erreurs révélées liés aux accès aux bases de données. Tsai et coll. expérimenté sur 60 services Web BBS avec 32 cas de test et dans ces

les expériences les cas de test négatifs ont révélé plus de défauts que les cas de test positifs.[21]

2. 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), et unsurvol sur les modèles formels qui permettent de décrire les spécifications formellement. Nous décrirons dans la suite quelquestypes,méthodes,architecturesetoutilsdetest.Enfinnousdétailleronsunepartiedesméthodesde test appliquées auxservices Web.

2.2- L'apparition des services web :

2.2.1 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 DevelopersConference 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.[5]

2.2.2 Définition des services web :

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. [5]

2.3- L'architecture orientée services (SOA) :

L'architecture orientée services (SOA) :

- Est une réponse efficace aux problèmes rencontrés par les organisations afin d'améliorer la flexibilité et réduire le coût de maintenance de leurs processus métiers.
- Est un concept et une approche pour développer des architectures distribuées centrées sur la notion de relation de service entre les applications et la formalisation de cette relation dans un contrat**.** [3]
- Est une réponse efficace aux problèmes rencontrés par les organisations afin d'améliorer la flexibilité et réduire le coût de maintenance de leurs processus métiers.

Actuellement, sous le vocable de SOA, se développe un style d'architecture orientée servicepermettantde construiredes systèmes informatiques évolutifs etadaptables, en améliorantleur qualité et en simplifiant leur intégration dans l'infrastructure informatique de l'entreprise,parrecours àdes composants réutilisables appelés services..[1]

Danslasuite,nousprésentonslesconcepts surlesquelsreposel'architectureSOA.

2.3.1 Définition service :

Un service est un logiciel autonome, auto-décrivant et modulaire qui exécute une fonction commerciale spécifique telle que la validation d'une carte de crédit ou la génération d'une facture

« ***Autonome***» implique des services comprenant tout ce qui est nécessaire pour les faire fonctionner.

« ***Auto-décrivant***» qu'ils ont des interfaces qui décrivent leurs fonctionnalités d'affaires.

« ***Modulaire*** » les services peuvent être agrégés pour former des applications plus complexes.

C'est l'application de principes de conception orientée services qui permettent de distinguer une unité en tant que service parmi d'autres unités qui peuvent exister en tant qu'objets ou composants **[3]**

Les services sont définis comme étant basés sur des standards et ils sont indépendants de la plate-forme et des protocoles afin d'assurer des interactions dans des environnements hétérogènes

Lorsqu'on parle de service, celui-ci est caractérisé par un point. Une interface, le contrat de service (le document WSDL) précise les messages d'entrée que peut recevoir ce point d'entrée pour réaliser la prestation.

En résumé, la notion de service est le produit d'une démarche d'abstraction par rapport au logiciel qui l'implémente, au processus qui l'exécute et au port qui le localise. Cependant, le service reste un objet très concret et technique.

La réalisation d'un service passe par des messages échangés, des transitions d'état et des actions que l'application cliente suppose assurés de la part de l'application prestataire du service**.** [4]

2.3.2 Composant d'une architecture orientée services :

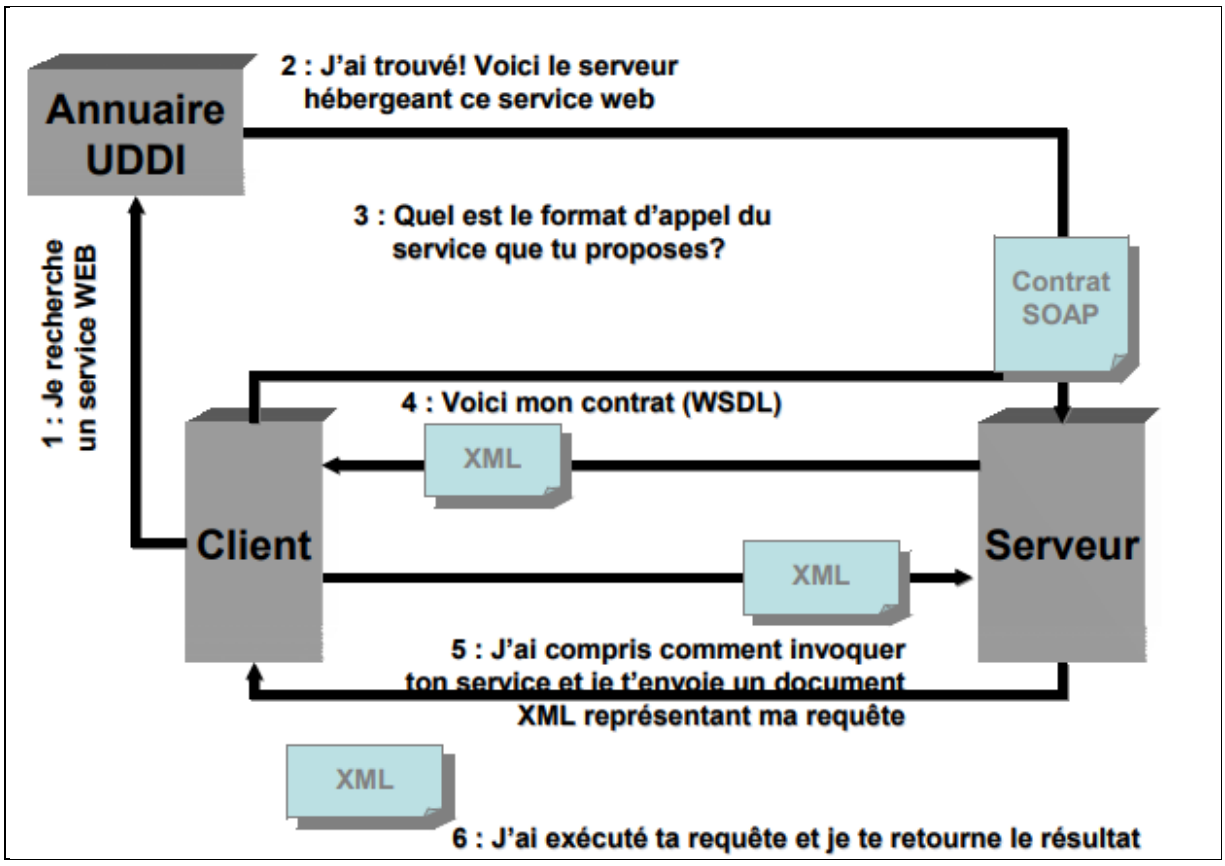


Figure 2-1 : Modèle fonctionnel de l'architecture SOA [4]

Décortiquonsceschéma:

- **Serviceproviderservice (serveur) :**

Le fournisseur de service met en application le service Web et le rend disponiblesurInternet.

- **Servicerequester (programmeclient) :**

C'est n'importe quel consommateur du service Web. Le demandeur utilise unserviceWebexistantenouvrantuneconnexionréseauetenenvoyantunedemandeenXML(REST, XML-RPC,SOAP).

- **Annuaireserviceregistry(UDDI) :**

registre de service est un annuaire de services. Le registre fournit un endroitcentraloùlesprogrammeurspeuventpublierdenouveauxservicesouentrouver.

Lesinteractions entrecestrois acteurs suiventplusieursétapes:

- **La publication du service** : le fournisseur diffuse les descriptions de sesservices Webdansl'annuaire.
- **Larechercheduservice** :leclientchercheunserviceparticulier,ils'adresse à un annuaire qui va lui fournir les descriptions et les URL desservicesdemandésafinde luipermettredelesinvoquer.
- **L'invocation du service** : une fois que le client récupère l'URL et ladescription du service, il les utilise pour l'invoquer auprès du fournisseurdeservices**.** [5]

2.3.3 Principes de conception orientée services :

Huit principes sont définis par Thomas Erl comme suit:

- ***Contrat de service standard***: Les services expriment leur but, leurs fonctionnalités, et leurs types de données via un contrat de service. Il adhère à un accord de communication standard, qui est défini collectivement par un ou plusieurs documents de description de service.
- ***Couplage faible du service*** (niveau de dépendance): Une relation spécifique à l'intérieur et à l'extérieur du service, en mettant l'accent sur la réduction des dépendances entre: le contrat de service, son implémentation et ses consommateurs .
- ***Abstraction de service***: les contrats de service ne contiennent que des informations essentielles et l'information sur les services est limitée à ce qui est publié dans les contrats de services.
- ***Réutilisation des services***: les services sont conçu pour être réutilisés. Ces services réutilisables sont conçus de manière à ce que leur solution logique soit indépendante d'un processus métier ou d'une technologie particulière.
- ***Autonomie des services***: les services exercent un niveau élevé de contrôle sur leur environnement d'exécution .

- ***Services sans états***: La gestion d'informations de l'état peut compromettre la disponibilité d'un service et compromettre son potentiel d'évolution. Les services sont donc idéalement conçus pour prendre un état sauf si cela est vraiment nécessaire.
- ***Possibilité de découverte de service***: les services sont complétés par des métadonnées communicatives par lesquelles ils peuvent être efficacement découverts et interprétés.
- ***Composabilité de service***: les services sont des participants efficaces de la composition, quelle que soit la taille et la complexité de la composition. [3]

Les services Web présentent l'implémentation la plus commune de SOA. Ceux-ci sont de plusen plus employés dans les entreprises pour construire des applications orientées Business ouWeb,enaméliorantleurqualitéetensimplifiantleurintégrationdansl'infrastructureinformatique del'entreprise.Quellessontlestechnologiesprometteusesutiliséesparlesservices Web?

2.4- Lestechnologies desservicesWeb :

Les technologies utilisées par les services Web sont HTTP,WSDL,REST,XML-RPC,SOAP et UDDI sur les services Web SOAP (appelés ci-après service (s) Web) en raison de leur popularité à la fois dans l'industrie et le monde universitaire.

2.4.1 SOAP :

SOAP(SimpleobjectAccessProtocol)estunprotocolestandarddecommunication. C'est l'épine dorsale du système d'interopérabilité. SOAP est unprotocole décrit en XML et standardisé par le W3C. Il se présente comme uneenveloppe pouvant être signée et pouvant contenir des données ou des piècesjointes.
Il circule sur le protocole HTTP et permet d'effectuer des appels de méthodes àdistance.[5]

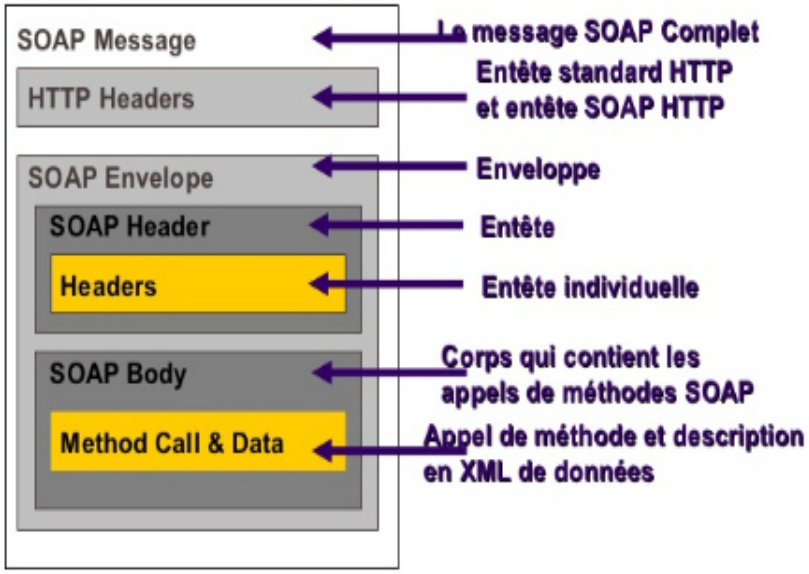


Figure 2-2:Structured'un messageSOAP[23]

2.4.2 XML-RPC :

XML-RPC est un protocole simple utilisant XML pour effectuer des messagesRPC. Les requêtes sont écrites en XML et envoyées via HTTP POST. Lesrequêtessontintégréesdanslecorpsdelaréponse HTTP.XML-RPCestindépendant de la plate-forme, ce qui lui permet de communiquer avec diversesapplications.Parexemple,unclientJavapeutparlerdeXML-RPCàunPerlServer. [5]

2.4.3 WSDL :

WSDL est un langage de description standard. C'est l'interface présentée auxutilisateurs.Ilindiquecommentutiliserleservice Webetcommentinteragiravec lui. WSDL est basé sur XML et permet de décrire de façon précise lesdétails concernant le service Web tels que les protocoles, les ports utilisés, lesopérations pouvant être effectuées, les formats des messages d'entrée et de sortieet lesexceptionspouvant être envoyées.[5]

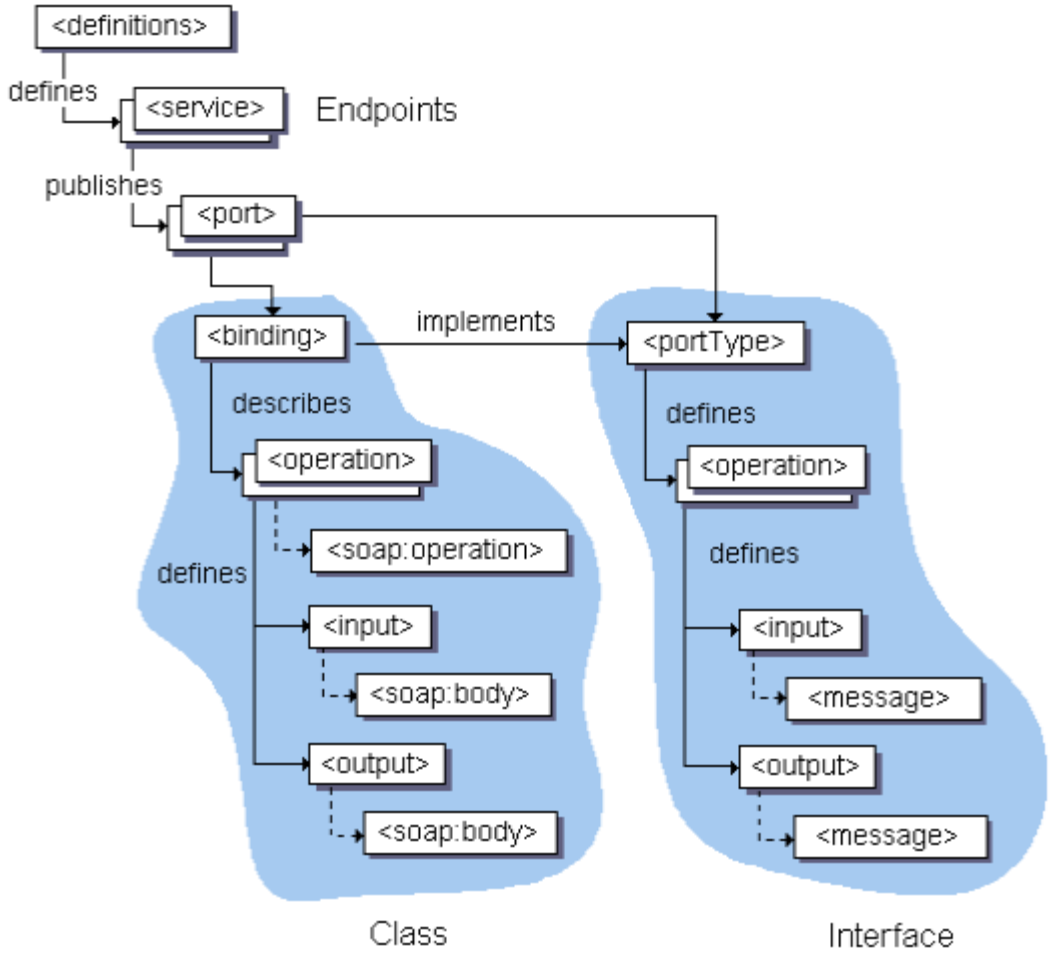


Figure 2-3: description WSDL [6]

2.4.4 UDDI :

UDDI (Universal Description, Discovery and Integration) est un annuaire deservices.Ilfournit l'infrastructure de base pour la publication etladécouvertedes services Web.

UDDI permet aux fournisseurs de présenter leurs servicesWeb auxclients.

Lesinformationsqu'ilcontientpeuventêtreséparéesentroistypes:

- lespagesblanches quiincluentl'adresse,lecontactetlesidentifiantsrelatifs auservice web.
- lespagesjaunesquiidentifientlessecteursd'affairesrelatifsauservice web.
- lespagesvertesquidonnentlesinformationstechniques.[5]

2.5- Les types de services web :

La majorité des grands sites Web (Amazon, eBay…) qui proposent des services Web auxdéveloppeurs,offrentsimultanémentdeuxcatégoriesdeservicesWeb: SOAPetREST.

2.5.1 Les services Web REST:

REST (Representational State Transfer): est une architecture de services Web. Élaborée en l'an 2000 par*RoyFiedling*, l'un des créateurs du protocole HTTP, du serveur Apache HTTPd etd'autrestravauxfondamentaux,RESTestunemanièredeconstruireuneapplicationpourles systèmes distribués commele World WideWeb.

REST exposent entièrement ces fonctionnalités comme un ensemble deressources(URI)identifiablesetaccessiblesparlasyntaxeetlasémantiqueduprotocole.[1]

2.5.2 Les services Web SOAP :

Appelés aussi WS-*exposent ces mêmes fonctionnalités sous laformedeservicesexécutables à distance.
Afin de rendre les services Web SOAP interopérables, l'organisation WS-I propose de définirles services Web en introduisant des profils, en particulier le profil WS-I Basic. Celui-ciest composé de quatre grandes parties:

- la description de l'interface du service Web grâce aulangageWSDL(WebServicesDescriptionLanguage).
- lasérialisationdesmessagestransmisvialeprotocoleSOAP(SimpleObjectAccessProtocol) .
- l'indexationdesservices Web dans des registres UDDI (Universal Description, Discovery Integration).

lasécuritédesservicesWeb,obtenueessentiellementgrâceàdesprotocolesd'authentificationet decryptageXML.[1]

2.5.3 Introduction XSD :

Un schéma XML décrit la structure d'un document XML.

Le langage XML Schema est également appelé XML SchemaDefinition (XSD).

Exemple XSD :

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>[7]
```

2.5.4 Le but d'un schéma XML :

Le but d'un schéma XML est de définir les blocs de construction juridiques d'un document XML :

- les éléments et attributs pouvant apparaître dans un document.
- le nombre (et l'ordre) des éléments enfants.
- types de données pour les éléments et les attributs.
- valeurs par défaut et fixes pour les éléments et les attributs. [7]

2.5.5 Pourquoi apprendre le schéma XML ?

Dans le monde XML, des centaines de formats XML standardisés sont utilisés quotidiennement,Bon nombre de ces normes XML sont définies par des schémas XML.

XML Schema est une alternative basée sur XML (et plus puissante) à la DTD.

- Les schémas XML prennent en charge les types de données

L'une des plus grandes forces des schémas XML est la prise en charge des types de données.

- Il est plus facile de décrire le contenu de document autorisé.
- Il est plus facile de valider l'exactitude des données.
- Il est plus facile de définir des facettes de données (restrictions sur les données).
- Il est plus facile de définir des modèles de données (formats de données).
- Il est plus facile de convertir des données entre différents types de données.

- Les schémas XML utilisent la **syntaxe XML.**

Une autre grande force des schémas XML est qu'ils sont écrits en XML.

- Vous n'avez pas besoin d'apprendre une nouvelle langue.
- Vous pouvez utiliser votre éditeur XML pour éditer vos fichiers Schema.
- Vous pouvez utiliser votre analyseur XML pour analyser vos fichiers Schema.
- Vous pouvez transformer votre schéma avec XSLT.

- Les schémas XML **sont extensibles**, car ils sont écrits en XML.

Avec une définition de schéma extensible, vous pouvez :

- Réutilisez votre schéma dans d'autres schémas.
- Créez vos propres types de données dérivés des types standard.
- Référencer plusieurs schémas dans le même document.

- Schémas XML Communication de données sécurisée.

Lors de l'envoi de données d'un expéditeur à un destinataire, il est essentiel que les deux parties aient les mêmes « attentes » concernant le contenu. [7]

Avec les schémas XML, l'expéditeur peut décrire les données d'une manière que le destinataire comprendra.

Exemple :

Une date comme : "03-11-2004" sera, dans certains pays, interprétée comme le 3 novembre et dans d'autres comme le 11 mars.

Cependant, un élément XML avec un type de données comme celui-ci :

<date type="date">2004-03-11</date>

assure une compréhension mutuelle du contenu, car le type de données XML "date" nécessite le format "AAAA-MM-JJ".

- Document XML Bien formé :

Un document XML bien formé est un document conforme aux règles de syntaxe XML, telles que :

- il doit commencer par la déclaration XML
- il doit avoir un élément racine unique
- les balises de début doivent avoir des balises de fin correspondantes
- les éléments sont sensibles à la casse.
- tous les éléments doivent être fermés.
- tous les éléments doivent être correctement imbriqués.
- toutes les valeurs d'attribut doivent être entre guillemets.
- les entités doivent être utilisées pour les caractères spéciaux. [7]

2.6- Qu'est-ce qu'une DTD ?

Une DTD est une définition de type de document, elle définit la structure et les éléments et attributs juridiques d'un document XML. [7]

2.6.1 Pourquoi utiliser une DTD ?

Avec une DTD, des groupes indépendants de personnes peuvent se mettre d'accord sur une DTD standard pour l'échange de données.

Une application peut utiliser une DTD pour vérifier que les données XML sont valides.

Une déclaration DTD interne

Si la DTD est déclarée dans le fichier XML, elle doit être encapsulée dans la définition <!DOCTYPE> :[7]

2.6.2 Document XML avec une DTD interne

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>[7]
```

Afficher le fichier XML :

Dans le fichier XML, sélectionnez « afficher la source » pour afficher la DTD.

- La DTD ci-dessus est interprétée comme ceci :

  - **!DOCTYPE note** définit que l'élément racine de ce document est note
  - **!ELEMENT note** définit que l'élément note doit contenir quatre éléments : "to,from,heading,body"
  - **!ELEMENT to** définit l'élément to comme étant de type "#PCDATA"
  - **!ELEMENT from** définit l'élément from comme étant de type "#PCDATA"
  - **L'en-tête !ELEMENT** définit l'élément d'en-tête comme étant de type "#PCDATA"
  - **!ELEMENT body** définit l'élément body comme étant de type "#PCDATA" [7]

2.7- Approchesdecompositiondesservicesweb:

Lacompositionpeutêtredécritesousdeuxangles:

2.7.1 Orchestration:

Uneorchestrationassemblelesserviceswebdansunprocessusmétierexécutablequi doitêtreexécutépar un moteurd'orchestration.

L'orchestrationdeservicesWeb(Figure 4)consisteàlaprogrammationd'unmoteurquiappelleunensembledesserviceswebselonunprocessusprédéfini.Cemoteurdéfinitleprocessusdanssonensembleetappellelesservices web (tant internes qu'externes à l'organisation) selon l'ordre des tâchesd'exécution.[5]

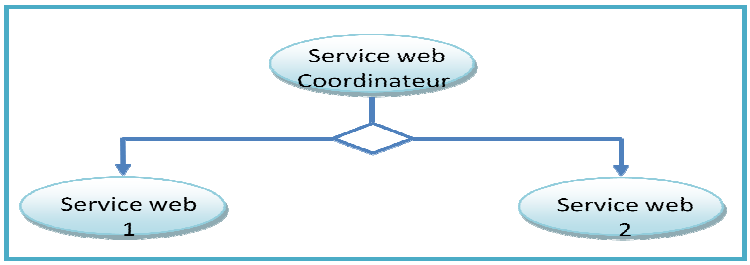


Figure 2-4: services web en architecture Orchestration [3]

2.7.2 Chorégraphie:

La chorégraphie n'implique pas un contrôle centralisé, le contrôle est partagéentre les participants en interaction. Une orchestration représente un processusexécutable à exécuter parunmoteurd'orchestrationenun seul endroit,alorsquelachorégraphieenessence représenteunedescription de lafaçondedistribuerlecontrôleentrelesparticipants collaborent, à l'aide desplusieurs moteurspour fairele travail.[5]

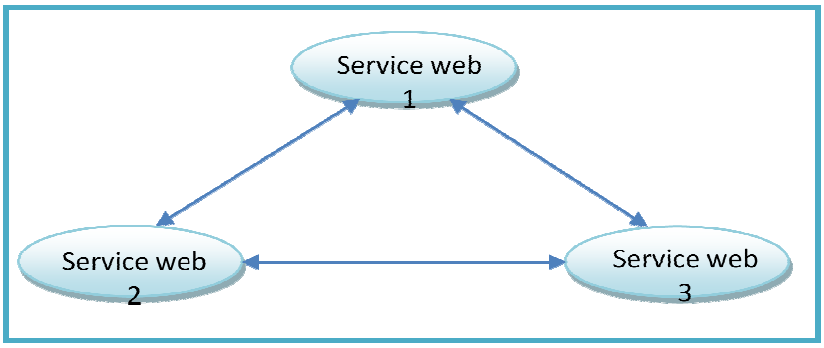


Figure 2-5:services web en architecture Chorégraphie [3]

2.8- Qu'est ce que le test ?

Si l'on se réfère aux fondamentaux, c'est-à-dire à la norme IEEE (Standard Glossary ofSoftwareEngineeringTerminology), le termede«**test**»apparaît comme:

« L'exécution ou l'évaluation d'un système ou d'un composant par des moyens automatiquesou manuels, pour vérifier qu'il répond à ses spécifications ou identifier les différences entrelesrésultats attendus etles résultats obtenus».

Pourpouvoirtesterunsystèmedonné,cedernier doitd'abordêtrespécifié par un modèle formel.

2.9- Description de modèle formelle grammaire d'arbre régulier (RTG) :

Pour les structurer les services web, nous avons besoin d'un modèle formel pour les documents XML.

Plusieurs modèles formels ont été créés [8.9.10.11] chacun pour répondre à un objectif différent. Jusqu'à présent, ces modèles sont marysyntaxique par nature et capturer des informations surstructure et types. Les modèles syntaxiques sont appropriés pour XML car il est rarement utilisé pour décrire le comportement.

Makoto [9] et Chidlovskii [10] donnent deux modèles similaires, que nous étendre pour les tests.

Notre modèle formel est une grammaire d'arbre régulier**- (RTG).** Nous utilisons RTG comme représentation formelle d'un schéma XML et dériver une arborescence de documents XML basée sur le RTG.[12]

2.9.1 Principe d'un modèle RTG :

Le principe d'un modèle grammaire arborescente régulière (RTG) est un 6-tuples<*E, D, N, A, P, n s* > , où:

1. ***E*** est un ensemble fini de types d'éléments.
2. ***D*** est un ensemble fini de types de données.
3. ***N*** est un ensemble fini de non-terminaux.
4. ***A*** est un ensemble fini de types d'attributs.
5. ***P*** est un ensemble fini de règles de production de deux formes:

• ***n → a <d>***, où ***n*** est non-terminal dans ***N, a*** est soit un type d'attribut en ***A,*** soit un type d'élément en ***E***, et ***d***est un type de données en ***D.***

• ***n → e <r>***, où ***n***est non terminal dans ***N, e***est un élément de ***E*** , et ***r*** est une expression régulière composé de non-terminaux.

6. ***n s*** est le non-terminal de départ, ***n s ∈N.***

Rm :Un modèle RTG peut être créé automatiquement à partir **d'un XML schéma ou DTD.** Étant donné une séquence de non-terminaux, nous pouvons remplacer à plusieurs reprises les non-terminaux par le côté droit des règles de production correspondantes.

Exemple : Considérez schéma les livres de l'exemple suivante il est capturé par un RTG ***G = <E, D, N, A, P, n $_s$>*** où:

- *E = {livres, livre, ISBN, prix, année}*
- *D = {chaîne, double, yearType}*
- *N = {n_s , n_b , n_i , n_r , n_y }*
- *A = {chaîne, double,yearType}*
- *P = { n_s → livres < (n b) *>*

  n b → livre <n_i n_r n_y>

  n_i → ISBN <chaîne>

  n_r → prix <double>

  n_y → année <yearType>}[12]

La figure 7 montre les étapes de dérivation pour le livre XML arbre document.

- La dérivation commence par le non borne n_s sur la figure 7.a.
- Les productions pour n_s et n_b sont représentés sur les figures 7.b et 7.c.
- Les productions pour n_i, n_r, et n_y sont sur la figure 7.d.**[12]**

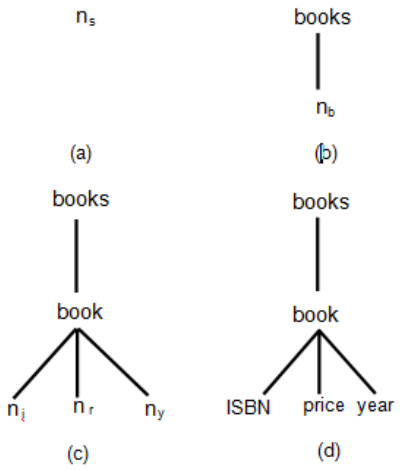


Figure 2-6 : XML Document TreeDerivation [12]

2.10- Techniques de test :

Cettesectionprésentequelquestechniques detest.

2.10.1 Test de partition :

Le test de partition est une technique de test qui vise à trouver des sous-ensembles de cas de test (à partir de cas de test existants) ,qui peut tester adéquatement un système. Le but du test de partition est de diviser le domaine d'entrée du SUT en sous-domaines, de sorte que sélectionner ou générer un certain nombre de cas de test à partir de chaque sous-domaine sera suffisant pour tester l'ensemble du domaine.[13]

2.10.2 Tests unitaires des services Web :

Les tests unitaires peuvent être considérés comme la technique de test la plus élémentaire et la plus naturelle applicable à tout système. Dans les tests unitaires, les unités individuelles d'un système qui peuvent être exécutées indépendamment sont considérées comme des unités.[13]

2.10.3 Test basé sur un modèle et vérification formelle des services Web :

Le test basé sur un modèle est une technique de test où les cas de test sont générés à l'aide d'un modèle qui décrit le comportement du SUT. Avantages des tests basés sur des modèles, tels que l'automatisation de la génération de cas de test processus et la capacité d'analyser la qualité du produit de manière statique en font une technique de test populaire.[13]

2.10.3.1 Test basé sur un modèle à l'aide de l'exécution symbolique :

L'exécution symbolique est utilisée comme base pour une technique de vérification qui se situe entre formel et informel vérification selon King. Dans les tests symboliques, le SUT est exécuté symboliquement à l'aide d'un ensemble de classes d'entrées au lieu d'un ensemble d'entrées de test.[13]

2.10.3.2 Test basé sur un modèle à l'aide de la vérification de modèle

La vérification par modèle est une méthode de vérification formelle et est décrite comme une technique de vérification d'un état fini systèmes concurrents selon Clarke et al. [28]. La vérification du modèle vérifie si le modèle du système peut satisfaire les propriétés données sous la forme d'une logique temporell[13]

2.10.4 Test basé sur un modèle à l'aide de Petri-Nets :

Petri-Net (Place / Transition Net) est une technique de modélisation mathématique et graphique permettant de spécifier et analyser des systèmes concurrents, asynchrones, distribués, parallèles, non déterministes et / ou stochastiques . Les Petri-Nets permettent différentes analyses sur le modèle telles que l'accessibilité, la délimitation, l'impasse, la vivacité,analyse de la réversibilité, de l'équité et de la conservation.[13]

2.10.5 Test basé sur le contrat des services Web

DbC [98] est une approche de développement logiciel où les contrats définissent les conditions (pré-conditions) pour un composant auquel accéder et les conditions (post-conditions) qui doivent

être maintenues après l'exécution de méthodes de ce composant avec les conditions préalables spécifiées. Utilisation de contrats, certains comportements inattendusdu SUT peut être détecté et les informations des contrats peuvent également être utilisées pour améliorer les tests processus lui-même.[13]

2.10.6 Test basé sur les pannes des services Web

Selon Morell , les tests basés sur les fautes visent à prouver que le SUT ne contient aucun défauts. La différence entre les cas de test basés sur les pannes et les cas de test réguliers est que les scénarios de test basés sur les erreursprouver l'inexistence de défauts connus plutôt que d'essayer de trouver des défauts qui existent. [13]

2.10.6.1 Perturbation XML / SOAP :

Les perturbations XML / SOAP sont effectuées en capturant les messages SOAP parmi les services et leurs utilisateurs.

Les messages défectueux sont générés à partir de ces messages capturés en injectant des défauts avant de les envoyer ou simplement en envoyant un message SOAP défectueux au service Web. Après des perturbations, le comportement du service Web avecle message défectueux est observé pour vérification.Un des premiers exemples de perturbation SOAP est proposé par Offutt et Xu .[13]

2.10.6.2 Injection de défaut au niveau du réseau :

L'injection de défaut au niveau du réseau (NLFI) est l'approche d'injection de défaut où les défauts sont injectés par ing, supprimer et réorganiser les packages réseau. Looker et coll. proposent l'utilisation de cette technique avec un framework appelé Web Service Fault Injection Tool (WS-FIT). Au niveau de la latence au niveau du réseau l'injection peut être effectuée avec une perturbation SOAP. WS-FIT peut effectuer les deux perturbations SOAPet les injections de latence. [13]

2.10.6.3 Mutation des spécifications du service Web :

Le test de mutation est une technique utilisée pour mesurer l'adéquation d'une suite de tests. Il est également utilisé dans le test génération de données et gestion des cas de test. Le test de mutation est effectué en injectant des défauts à SUT tels que changer et modifier le code source ou changer l'état du SUT pendant l'exécution du test. Ces changementssont appliqués en utilisant les règles de transformation appelées opérateurs de mutation.[13]

2.10.7 Tests collaboratifs :

Le test logiciel collaboratif est le concept de test où plusieurs parties impliquées dans un service Web, comme en tant que développeur, intégrateur, testeur et utilisateur, participer au processus de test. Les tests collaboratifs sont généralement utilisés dans les techniques de test telles que les étapes d'utilisabilité où la fonctionnalité correcte est testée avec participationde différentes parties. [13]

2.10.8 Test de régression des services Web :

Les tests de régression sont la réutilisation des cas de test existants des tests système précédents. Les tests de régression est effectuée lorsque des ajouts ou des modifications sont apportés à un système existant. Dans la régression traditionnelle test, on suppose que le testeur a accès au code source et le test de régression est effectué dans un manière boîte blanche. La réalisation de tests de régression en boîte blanche aide principalement à la gestion des cas de test. [13]

2.10.9 Test d'interopérabilité des services Web :

L'interopérabilité est la capacité de plusieurs composants à travailler ensemble, c'est-à-dire à échanger des informations et traiter les informations échangées. L'interopérabilité est un problème très important dans les plates formes ouvertes comme SOA. Même si les services Web doivent être conformes aux protocoles standard et aux spécifications de service,des problèmes d'incompatibilité peuvent encore survenir.

Le besoin d'interopérabilité entre les spécifications de service est reconnu par l'industrie et WS-I, uneorganisation industrielle ouverte, formée par les principales sociétés informatiques. L'organisation a défini un WS-I Basic Profile afin de garantir l'interopérabilité des services Web. L'organisation WS-I assure l'interopérabilité scénarios qui doivent être testés et un certain nombre d'outils pour aider le processus de test. [13]

2.10.10 Test d'intégration des services Web

Il est important d'effectuer des tests d'intégration en génie logiciel pour s'assurer que tous les composants fonctionnent comme un système. Puisque l'idée derrière SOA est d'avoir plusieurs services faiblement couplés et interopérables pour former un système logiciel, les tests d'intégration dans SOA deviennent une nécessité. En effectuant des tests d'intégration,tous les éléments de SOA peuvent être testés, y compris les services, les messages, les interfaces et les processus métier .[13]

2.11- Outils de test :

2.11.1 SoapUI :

SOAPUI est un outil de test d'API de premier plan pour tester les services Web.

Il peut vérifier à la fois les services Web SOAP et les services Web RESTful.

SoapUI est disponible en version open-source et PRO, mais comme vous pouvez le deviner, la version PRO a des fonctionnalités supplémentaires. Il est basé sur Java, donc il fonctionne sur la majorité des systèmes d'exploitation; avant tout, il est facile à apprendre et à utiliser et fiable pour tous.

Approuvés par des millions d'utilisateurs, vous pouvez les utiliser pour divers cas de test.

- Fonctionnel.
- Axée sur les données.
- Railleur.
- Sécurité.
- Performance.[24]

2.11.2 Repos assuré :

Repos assuré Library est un outil API sur mesure pour le domaine Java qui utilise des personnes pour tester et valider les services REST. Il est également utilisé pour tester les services Web

HTTP, JSON et XML, et il nous offre de nombreuses fonctionnalités clés telles que la validation XPath, la syntaxe du chemin JSON, les téléchargements de fichiers faciles et la réutilisation des spécifications. Et aussi, il a été principalement influencé par des langages dynamiques tels que Groovy, Ruby.

L'une des excellentes fonctionnalités offertes par RestAssured est que vous n'avez pas besoin d'analyser les réponses XML ou JSON après avoir obtenu la réponse.[24]

Le tableau suivant illustre d'autres outils de test les services web avec leur caractéristique

l'outil	Plate-forme	À propos de l'outil	Meilleur pour	Prix
ReadyAPI	Windows, Mac, Linux.	Il s'agit de la plate-forme pour les tests fonctionnels, de sécurité et de charge de RESTful, SOAP, GraphQL et d'autres services Web.	Tests fonctionnels, de sécurité et de charge des API et des services Web.	Cela commence à 659 $ / an.
Catalogue Studio	Windows, macOS, Linux	Un outil complet de test API, Web, Desktop et Mobile pour les débutants et les experts.	Test automatisé	Licence gratuite avec services d'assistance payants

Tableau 2-1:les outils de test [24]

2.12- Classification de test

Plusieurs classifications de tests ont présenté. Nous détaillons dans cette section celle proposéeparJ. Tretmans selontrois axes. Cetteclassification est illustréedanslaFigure2-7.

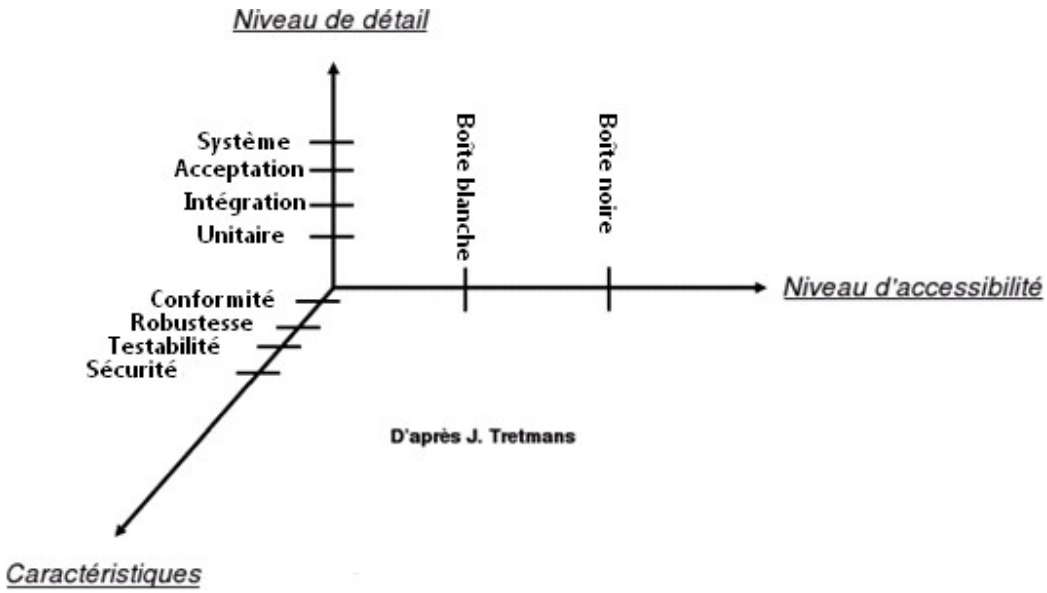


Figure 2-7:Classificationdetests(Tretmans)[1]

Dansle**niveaudedétail**,quatreniveauxdetestse présentent:

– testunitaire,quivérifie lesfonctionsuneparune, enisolationdurestedu système,

– testd'intégration,quivérifielebon enchaînementdesinteractionsentresystèmesintégrés,

– testsystème,quivérifielecomportement globaldusystèmelorsdelaphased'assemblage,

–testd'acceptation,quipermetdedéterminersiunsystèmesatisfaitounondescritèresd'acceptation.[1]

Le**niveaud'accessibilité**estregroupéendeux catégories:

- **les tests boîte blanche** : qui sont applicables sur des systèmes dont la structure interne estconnue. Par exemple, les boucles d'un code source peuvent être testées (les initialisations, lesconditionsetc.). Cetypede test estappeléaussi test structurel,
- **les tests boîte noire** : qui sont utilisés sur des systèmes dont la structure interne n'est pasconnue. Par conséquent, les testeurs ne peuvent avoir accès qu'aux interfaces. Pour vérifierqu'une opération fonctionne correctement, le test s'appui sur les informations retournées parcelle-ci.

Nous notons qu'une combinaison des deux approches précédentes est appelée: tests en boîtegrise.

Le troisième axe expose des caractéristiques à tester:larobustesse, laconformité,latestabilitéet lasécurité. [1]

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. 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, spécification d'un système par un modèleformel, les différents technique, outil ,classification de test existant L'étude détaillé l'un des technique de test seront décrites dans le chapitre suivant.

3. Introduction :

Les tests sont l'une des phases les plus importantes du développement de tout produit ou logiciel. Il existe différents types de tests logiciels qui doivent être effectués pour répondre aux besoins du logiciel, La perturbation des données sur les valeurs et les interactions. L'ensemble de ces messages modifiés est ensuite utilisé comme suite de tests. Cette mémoire étend la technique de test de perturbation des données par Offutt et Xu .

3.1- Perturbation des données :

La perturbation des données (DP) est notre principale méthode de test Composants de service Web.

Le processus fonctionne en modifiant messages de demande, renvoi des messages de demande modifiés, et analyser les messages de réponse pour un comportement correct.

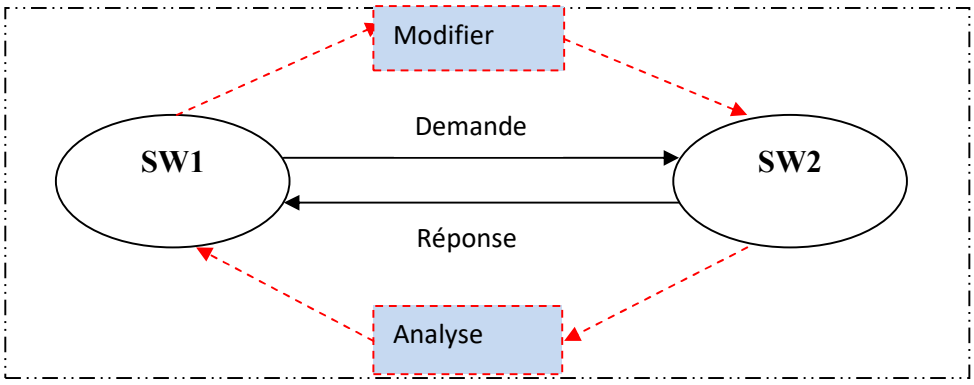


Figure 0-1 : Principe de perturbation

Les messages qui fournissent des appels RPC traditionnels à l'aide de SOAP sont dits être des *communications RPC*.

Les services Web sont certaines fois les programmes qui transfèrent des données d'un endroit à l'autre,

Par exemple, les servlets Java. Les messages dont le but est de transmettre Les données sont appelées *communications de données*.

Dans la communication de données les fichiers WSDL ne sont pas suffisants pour les tests car les fichiers WSDL ne contiennent pas d'informations sur le format de message.

La plupart des outils de test de services Web qui testent les messages SOAP se concentrent sur le test des communications RPC et ne inclure la communication de données.
Cette recherche utilise des données perturbation pour tester les deux types de communications.
Pour ce faire, la perturbation des données inclut à la fois la valeur des données perturbation et perturbation d'interaction.

- ***Perturbation valeur des données*** : modifie les valeurs des messages SOAP en termes de types de données.
- ***La perturbation d'interaction*** *:* modifie les messages en RPC et communications de données.

RPC :

En informatique eten télécommunication, **RPC** (***r**emote **p**rocedure **c**all*)estun protocole réseau permettant de faire des appels de procédures sur un ordinateur distant à l'aide d'un serveur d'applications. Ce protocole est utilisé dans le modèle client-serveur pour assurer la communication entre le client, le serveur et d'éventuels intermédiaires. [22]
Il existe des différences entre les **communications RPC** et les **communications de données** qui affectent les aspects pratiques des tests.
Principalement, la méthode nécessaire pour **extraire les données d'un message** est différente.
Les valeurs dans *communications RPC* sont directement extraites par les fonctions de procédure distantes. Les programmeurs n'ont pas besoin d'analyser les messages.

Les valeurs en *communications de données* doivent être analysées par le programme.

Une autre différence réside dans **les relations entre message** éléments.

Les messages RPC ont un format plat. En termes de l'arborescence, les éléments de données RPC sont tous frères, alors que la communication de données utilise souvent XML, ce qui permet une structure hiérarchique plus compliquée parmi les éléments.

Les messages de communication de données peuvent inclure des informations sur les relations et les contraintes des entités de données.

Données les types de messages de communication de données sont spécifiés dans Schémas XML que deux services Web acceptent de suivre.

Lorsque deux services Web utilisent des communications de données, ils doivent avoir un accord sur le format du message, qui est généralement codé dans un schéma XML ou DTD. [12]

Les trois sous-sections suivantes décrivent des variantes de la perturbation.

- DVP : perturbation de la valeur des données modifie les valeurs.
- RPC : perturbation de la communication test les données utilisent.
- CRP : perturbation communication de donnes teste les relations et les contraintes sur les données.

3.1.1 Perturbation de la valeur des données (DVP)

La perturbation de la valeur des données (DVP) modifie les valeurs dans SOAP messages selon les règles définies sur les types de Valeur.

La perturbation de la valeur des données repose sur les idées de la frontière test de valeur .

Recommandation de type de données XML du W3C répertorie 19 types de données primitifs.

Mais en raison des limites de temps, nous ne listons que les règles pour les cinq types qui correspondent aux types primitifs dans la plupart langages de programmation.

Type de données	Valeurslimites
String	Maximum length, minimum length, uppercase,lowercase
Numeric	Maximumvalue, minimumvalue, zero
Boolean	true,false

Tableau 3-1: Valeurs limites pour la Perturbation valeur des données [12]

Les cas de test sont dérivés de la valeur par défaut :

- valeurs limites des schémas XML.
- Les valeurs limites pour numérique (décimal, flottant et double), chaîne et booléen.
- les types sont répertoriés dans le tableau 3.
- Les tests sont créés en remplaçant ***chaque valeur*** avec ***chaque valeur limite***, à son tour, pour la valeur approprié type .

Dans le tableau 3, la ***longueur maximale*** et ***minimale*** des chaînes.

- sont la longueur maximale et minimale autorisée des chaînes dans les schémas XML.
- Les valeurs ***majuscules*** et ***minuscules*** sont en interprétés comme des opérateurs qui définissent la casse pour chaque lettre dans la chaîne.
- La *valeur maximale* et *minimale* pour numérique les valeurs sont les valeurs maximales et minimales autorisées dans Schémas XML.
- Les opérateurs *zéro* modifient les valeurs numériques à zéro.
- *Le* type *booléen* a deux valeurs: true et false.
- Le document XML pour un livre des exemples 1 et 2 utilise trois types de données: string, int et double.

Certains des tests DVP les cas qui seraient créés sont indiqués dans le tableau 3 (en raison de restrictions d'espace, seules les valeurs représentatives sont affichées, pas tout).

- La chaîne «_» représente une valeur nulle.

3.1.2 Perturbation de la communication RPC :

Les services Web sont souvent utilisés pour simplifier la procédure à distance appels (RPC) en utilisant SOAP pour formater et transmettre les appels.

Les messages RPC incluent des valeurs pour les arguments des fonctions de procédure à distance.

Comme dit précédemment, Les éléments dans les messages RPC sont des frères et les valeurs de ces éléments sont directement extraits par les fonctions. Les messages de réponse sont des résultats calculés par la fonction perturbation de la communication RPC se concentre sur les tests les données utilisent, divisées en deux types: les utilisations normales des données et utilisation SQL.

Les utilisations normales des données se produisent lorsque des ***valeurs*** sont utilisées dans les programmes.

Les utilisations SQL se produisent lorsque des ***chaînes*** sont utilisées comme entrées dans une base de données.

L'analyse de mutation est une technique de test de logiciel qui crée des versions modifiées de programmes, appelées mutants, et demande ensuite au testeur de trouver des entrées de test pour provoquer le **mutant** les programmes échouent.

Cette recherche utilise le même concept, apportant de petites modifications aux objets syntaxiques, mais l'applique à modifier les valeurs au lieu des programmes.

Cela constitue un major avance et divergence dans l'application de la mutation Analyse.

La perturbation des données est utilisée pour générer des valeurs liées aux utilisations normales des données, et l'injection SQL est étendue pour tester les utilisations SQL.

Les opérateurs de mutation traditionnels modifient code source, pas de données, donc différents types d'opérateurs ont été défini.

Valeurd'origine	Valeursperturbées	Cas de test
<ISBN>0-672-32374-5</ISBN>	00000000000000000000000000	maximumlength
	—	minimumlength
<price>59.99</price>	$2^{63}-1$	maximumvalue
	-2^{63}	minimumvalue
	0	Zero
<year>2002</year>	$2^{32}-1$	maximumvalue
	-2^{32}	minimumvalue
	0	Zero

Tableau 0-2: Test de document de livre pour la perturbation de la valeur des données [12]

Tout d'abord, nous présentons une définition abstraite d'une mutation opération qui peut s'appliquer à la fois à la mutation du texte du programme et perturbation des données. Formellement, un opérateur de mutation abstrait est défini comme :

Définition 1 :

Étant donné un ensemble de toutes les instances d'élément **N**, a l'opérateur de mutation est **r = *f* (n_1 , ..., n_i)** , où ***f*** est une fonction , **i ≥ 1** , chaque **n_1 , ..., n_i ∈ N** et a le même type de données, et **r** a le même type de données que n_1 , ..., n_i .

Les opérateurs de type de données numériques ont été définis aux tableaux suivants:

Nom de l'opérateur	**Brève description**
Diviser (*n*)	Changer la valeur ***n*** par **1 ÷ *n*** , où ***n*** est un double type de données.
Multiplier (*n*)	Changer la valeur ***n*** en ***n* * *n***
Négatif (*n*)	Changer la valeur ***n*** en **- *n***
Absolu (*n*)	Changer la valeur ***n*** en **| *n*|**
Échange (n_1, n_2)	Substituer la valeur n_1 pour n_2 et vice versa, où n_1 et n_2 ont le même type.
Non autorisé (*str*)	Changer la valeur de chaîne ***str*** par ***str***'OR' 1 '=' 1

Tableau 0-3 : les operateurs numérique

Lorsqu'un service Web a deux arguments du même type, alors peut être échangé.
Une ***injection SQL*** se produit lorsqu'un l'instruction SQL est incluse dans une valeur de chaîne qui est soumise connecté à une application ou un service Web.

L'attente est que la chaîne sera stockée dans une base de données, et le SQL étranger l'instruction sera exécutée.

Cela peut permettre à des accès aux bases de données sur le serveur, avec une potentielle importante conséquence pour le serveur et la base de données.

Décrit plusieurs techniques courantes d'injection SQL .

Nous étendons ***autorisation de contournement*** aux utilisations SQL et définissez le avec l'opération ***Non autorisé () .***

Exemple *:* Un message SOAP pour la demande de connexion

<? xml version = "1.0" encoding = "UTF-8"?>

<soapenv: enveloppe

xmlns: xsd = "http://www.w3.org/2001/XMLSchema"

xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance">

<soapenv: Corps>

<adminLoginsoapenv: encodingStyle =

"http://schemas.xmlsoap.org/soap/encoding/">

<arg0 xsi: type = "xsd: string">bahia</arg0>

<arg1 xsi: type = "xsd: string">abcd</arg1>

</adminLogin>

</ soapenv: Corps>

</ soapenv: Enveloppe>

L' opérateur ***Unauthorized ()*** modifie l'exemple dans le manière suivante:

Exemple *:* Le corps de connexion SOAP modifié Un message

.....

<soapenv: Corps>

<adminLoginsoapenv: encodingStyle =

"http://schemas.xmlsoap.org/soap/encoding/">

<arg0 xsi: type = "xsd: string">**bahia** 'OR' **1 '=' 1**</arg0>

<arg1 xsi: type = "xsd: string">**abcd**'OR' **1 '=' 1**</arg1>

</adminLogin>

</ soapenv: Corps>

.....

- La requête SQL d'origine ressemble à ceci:

  SELECT le nom d'utilisateur FROM adminuser WHERE username = 'bahia' ET mot de passe = 'abcd' .

- Une fois l'opérateur ***Unauthorized ()*** utilisé, la requête SQL devient:

  SELECT le nom d'utilisateur FROM adminuser WHERE username = 'bahia' OU '1' = '1' ET mot de passe = 'abcd' OU '1' = '1'

- La première requête SQL sélectionne uniquement l'enregistrement où le nom d'utilisateur = "bahia" et mot de passe = "abcd", et renvoie une valeur nulle si cette combinaison n'est pas dans la base de données.
- La deuxième La requête SQL sélectionne tous les noms d'utilisateur de la table administrateur

3.1.3 Perturbation de la communication de données (DCP) :

La plupart des outils de test pour les messages SOAP se concentrent sur le test de RPC Communications et n'incluent pas la communication de données.

Cependant, la communication de données est une partie importante du Web l'interaction des services et doit être testée.

A titre d'exemple, considérer une librairie en ligne.

- Le serveur Web du livre store, recueille des informations pour la commande d'un client en XML format,
- organise le document XML au format SOAP.
- envoie le message SOAP aux serveurs des éditeurs de livres.

Le document XML comprend des informations sur le Customer et l'ordre.

Le but de la communication de données est pour transférer des données, il comprend donc généralement plus de données que RPC.

Les messages de communication le font. Par exemple, il est courant pour inclure les relations et les contraintes de base de données.

Par conséquent, ***la perturbation de la communication de données se concentre sur les tests ces relations et contraintes***. Cependant, la sémantique de ces relations et contraintes sont légèrement différentes à partir de ceux de la littérature de la base de données.

Pour gérer cette différence, nous utilisons le modèle formel ***RTG*** pour définir les messages. Les relations et les contraintes sont incluses dans l'ensemble fini de règles de production ***p***dans le RTG.

La règle de production $n \rightarrow a\langle d \rangle$ est une règle de production à partir d'un élément non terminal à un élément terminal ou à un attribut.

$n \rightarrow e\langle r \rangle$ est une règle de production à partir d'un élément non terminal à un autre élément non terminal. Formellement, la relation et les contraintes peuvent être définies avec RTG comme suit:

Définition 2 :

Étant donné un schéma XML G = <E, D, N,A, P, n_s>, une relation est une règle de production dans **P: n →e <r>,** où **n** est un non-terminal dans **N**, **e** est un élément dans **E**, et **r** est une expression régulière composée de non-terminaux.

Définition 3 :

Étant donné un schéma XML G = <E, D, N,A, P, n_s>, une contrainte est une règle de production dans **P: n → β <τ>,** où **n** est un non terminal dans **N**, **β** est un élément dans **E ,**ou un attribut dans **A**, et **τ** est un type de données dans **D**.

Pour les relations, nous nous concentrons sur le test des données référentielles l'intégrité, qui est l'ensemble des règles qui régissent les relations entre les **clés primaires** et les **clésétrangères** des tables au sein d'un base de données relationnelle. Ils sont utilisés pour déterminer la consistance.

Les schémas XML définissent les relations comme parent-enfant associations entre deux éléments non terminaux.

Un schéma XML utilise ***maxOccurs***pour spécifier des relations référentielles entre les éléments parents et les éléments enfants.

Dans une régulière expression d'une relation, les opérateurs '**?**', '**+**' et '*****' désignent **zéro** ou **un,au moins un** et **n'importe quel nombre d'élément occurrences**. Ces opérateurs reflètent les contraintes de cardinalité dans un schéma XML.

Les stratégies sont définies comme suit:

• Étant donné une relation $n \rightarrow e\langle r \rangle$, s'il y a une expression α **?** dans ***r***, il y aura deux cas de test. L'un en contient **un** α et l'autre contient une **instance vide.**

• Étant donné une relation $n \rightarrow e\langle r \rangle$, s'il y a une expression α + dans ***r*** , il y aura deux cas de test. L'un en contient **un** α et l'autre contient un nombre **d'instances** α .

• Étant donné une relation $n \rightarrow e\langle r \rangle$, s'il y a une expression $\alpha *$ dans ***r*** , il y aura deux cas de test. On contient $\alpha * \alpha$et l'autre contient $\alpha *^{-1}$.

où $\alpha * \alpha$ duplique une instance d'élément et $\alpha *^{-1}$ supprime un élément dans la position.

Le tableau suivant illustre ces points :

Expression régulière $n \rightarrow e(r)$	Cas de test	Off & Xu	De nouvelles techniques
α ? en *r*	α instance	✓	*
	Vide	✓	*
α + en *r*	α instance	✓	*
	α instance	✓	*
α* en *r*	α*α	✓	*
	α^{*-1}	✓	*

Tableau 3-4 : Opérateurs de mutation - données. [25]

Les contraintes de base de données sont des règles qui régissent les expressions d'attributs dans les tables d'une base de données relationnelle. Par exemple, *nulle non autorisé* et *l'unicité non nulle* est souvent utilisé dans une base de données.

Dans les schémas XML, la définition les contraintes est légèrement différente.

Une contrainte dans un schéma XML est une association parent-enfant entre un élément non terminal et un élément terminal.

Toutes les contraintes de composant définies dans la recommandation de type de données XML du W3C sont considérés le schéma XML pour les livres. L'ensemble de production *P* contient trois contraintes:

- $n_i \rightarrow$ *ISBN<chaîne>*,
- $n_r \rightarrow$ *prix<double>*,
- $n_y \rightarrow$ *année<annéeType>*.

les deux premières règles de production n'ont aucune contrainte de composant.

La contrainte de composant pour la troisième règle de production est

<*xs* : ***totalDigitsvalue***= "4">. Le cas de test créé pour cette contrainte de composant est 9999.

3.2- Principe de Perturbation XML / SOAP :

Les perturbations XML / SOAP sont effectuées en capturant les messages SOAP parmi les services et leurs utilisateurs.
Les messages défectueux sont générés à partir de ces messages capturés en injectant des défauts avant de les envoyer, ou simplement en envoyant un message SOAP défectueux au service Web. Après des perturbations, le comportement du service Web avec le message défectueux est observé pour vérification.
Un des premiers exemples de perturbation SOAP est proposé par Offutt et Xu .

Offutt et Xu proposer trois types de perturbations différents :

Perturbation de la valeur des données (DVP) :qui est effectuée en modifiant les valeurs dans un message SOAP.

*Interférences de communication des appels de procédure distante (RCP) :*qui sont effectuées en modifiant les arguments des procédures distantes.[12]

Offutt et Xu proposent l'application de l'analyse de mutation à objets syntaxiques et perturbation des données avec le code SQL.
La perturbation du code SQL facilite également SQL test d'injection.
Perturbations de communication de données (DCP) :utilisées pour tester les messages qui incluent la base de données relations et contraintes. [12]

3.3- Les principaux facteurs qui contribuent aux difficultés de test :

Les services Web comprennent:

- Les différences de développement et les environnements d'application, qui augmentent les tests difficulté de fonctionnement réel avant les services Web sont déployés.
- Utilisation de méthodes de test automatiques et techniques, qui nécessitent une interface de service pour la conception et mise en œuvre.

- Les performances et l'évolutivité des services Web en test, qui doivent être pris en compte lorsqu'un grand nombre d'utilisateurs accède simultanément à un serveur vice à travers différents environnements.
- Services Web des caractéristiques qui diffèrent des objets de test réguliers, tels que distribution, découverte dynamique, liaison dynamique et invisibilité invoquant des processus.
- L'augmentation du risque de sécurité après la publication des méthodes de service et des interfaces, qui augmente la possibilité d'attaquer le système.
- L'application de services Web implique généralement un service fournisseurs, éditeurs et utilisateurs, qui doivent tous être impliqués à différentes étapes du test.[25]

3.4- Les approches de test :

Le tableau suivant illustre les différents types d'approche de test existant avec leur auteur et les technologies utilisant.

Author	TestingApproach	WSTechnology	Tools	ExperimentedWeb Services
Bartolinietal.	Generation Specification based Test data	WSDL	WS-TAXI	Picoservice
Maetal.	Generation Specification based Test data	WSDL	–	Syntheticordersystem(3versions)
Baietal.	Generation Specification based Test data	WSDL	–	356publicservices
Lietal.	Generation Specification based Test data	WSDL	WSTD-Gen	Callcenterqueryservice
Paradkaretal. DongandYu	Model-basedtesting Model-basedtesting	OWL-S WSDL	– –	Anonlineapplicationforasset management –
Maetal.	Model-basedtesting	BPEL	–	–
Guangquanetal.	Model-basedtesting	BPEL	–	Bookrenewalprocess
Conroyetal.	Testdatageneration	WSDL	–	–

Yanetal.	Model-basedtesting	BPEL	–	Loanapproval
Yuanetal.	Model-basedtesting	BPEL	–	–
Endoetal. KaschnerandLohmann	Model-basedtesting Model-basedtesting	BPEL BPEL	ValiBPEL –	GCD,Loanapprovalandnicejourney Exampleonlineshop
Houetal. Blancoetal.	Testcasegeneration Testcasegenerationusing	BPEL BPEL	– –	Loanapproval(1-2),atm,marketplace,gymlocker,BPEL(1-5) Loanapproval
Bertolinoetal.	search-basedmethods Partitiontesting	WSDL	TAXI	–
Baietal.	Partitiontesting	OWL-S	–	Syntheticravelsystem
SneedandHuang Lenzetal.	Unittesting Model-drivenunittesting	WSDL –	WSDL Test –	eGoverment project with9 WebServices(WSs) –
Chanetal. Tsaietal.	Unit testing using metamorhpicrelations Unittesting	– WSDL	– ASTRAR	– –
MayerandLübke	Unittesting	BPEL	BPELUnit	–
Lietal.	Unittesting	BPEL	–	–
SinhaandParadkar	Model-basedtesting	WSDL-S	–	–
Fuetal.	Model-checking	BPEL	SPIN	–

Garcıa-Fanjuletal.	Model-basedtesting	BPEL	SPIN	Loanapprovalexample
Zhengetal.	Model-basedtesting	BPEL	SPIN,NuSMV	–
Huangetal.	Model-basedtesting	OWL-S	BLAST	SyntheticshoppingWS
Betin-CanandBultan	Model-checking	BPEL	JavaPathfinder, SPIN	Synthetictravelagencyservice
RamsokulandSowmya	Model-checking	WSprotocols	SPIN	WS-AT
Lallaietal.	Model-basedtesting	BPEL	BPEL2IF,TestGen-IF	OracleexampleloanService
DongandYu	Model-basedtesting	WSDL	–	–
Wangetal.	Model-basedtesting	OWL-S	TCGen4WS	–

Tableau 0-5:les approches de test [21]

Conclusion :

Les tests basés sur les fautes visent à prouver que le service sous test ne contient aucuns défauts. La différence entre les cas de test basés sur les pannes et les cas de test réguliers est que les scénarios de test basés sur les erreurs prouver l'inexistence de défauts connus plutôt que d'essayer de trouver des défauts qui existent.

Les tests basés sur les pannes peuvent tester les services Web pour les pannes courantes qui peuvent exister dans l'environnement SOA et augmenter la fiabilité des services.

4. Introduction :

A travers ce chapitre, nous avons d´écrit les différents composants pour implémenter notre outil dans un système des comptes bancaire. nous allons définir les objectifs (résultats attendus) et les fonctionnalités des acteurs, puis nous avons présenté la conception globale, l'architecture de notre système, et la conception détaillée du système,ainsi nousavonsdécrirelamiseenœuvredesdifférentesétapesdenotre systèmeconçudanslechapitreprécédent.Ils'agiticid'expliquerl'environnementmatérieletlogiciels urlequelnotresystèmeaétédéveloppé.Nousavonscommencéparlajustificationdel'environnementde développementutiliséainsiparlesoutilsetleslangages deprogrammationsutilisés,ensuitelaplate-formechoisieetenfinprésentélesapplicationsnécessairesà l'implémentationdenotresystème.

4.1- Conceptiondétaillée

Nousallonsdécrirelesfonctionnalitésdenotreconceptionetlesdifférentsdiagrammes essentiels.

4.1.1 Lediagrammedecasd'utilisation (modélisationfonctionnelle) :

Lediagrammedecasd'utilisationduclientavecleursfonctionnalitéestprésentécommesuit:(figure4-1)

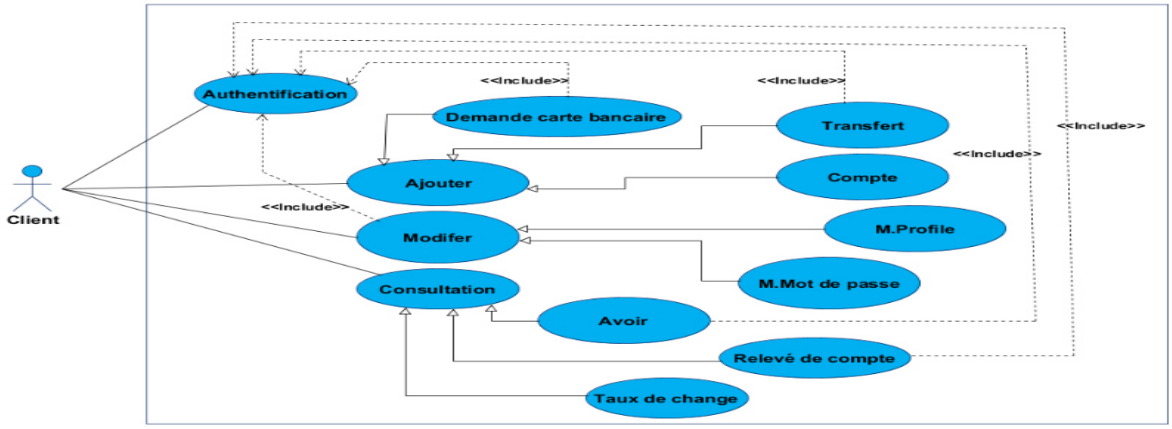


Figure 4-1: Diagrammedecasd'utilisation

4.1.2 Lediagrammedeclasse(Modélisationstatique) :

Enutilisantlediagrammedeclassesquireprésentelesentités(classes)statiquesdans l'application,Notreconceptionestreprésentécommesuit(figure4-2)Lerôledechaque classestcommesuit:

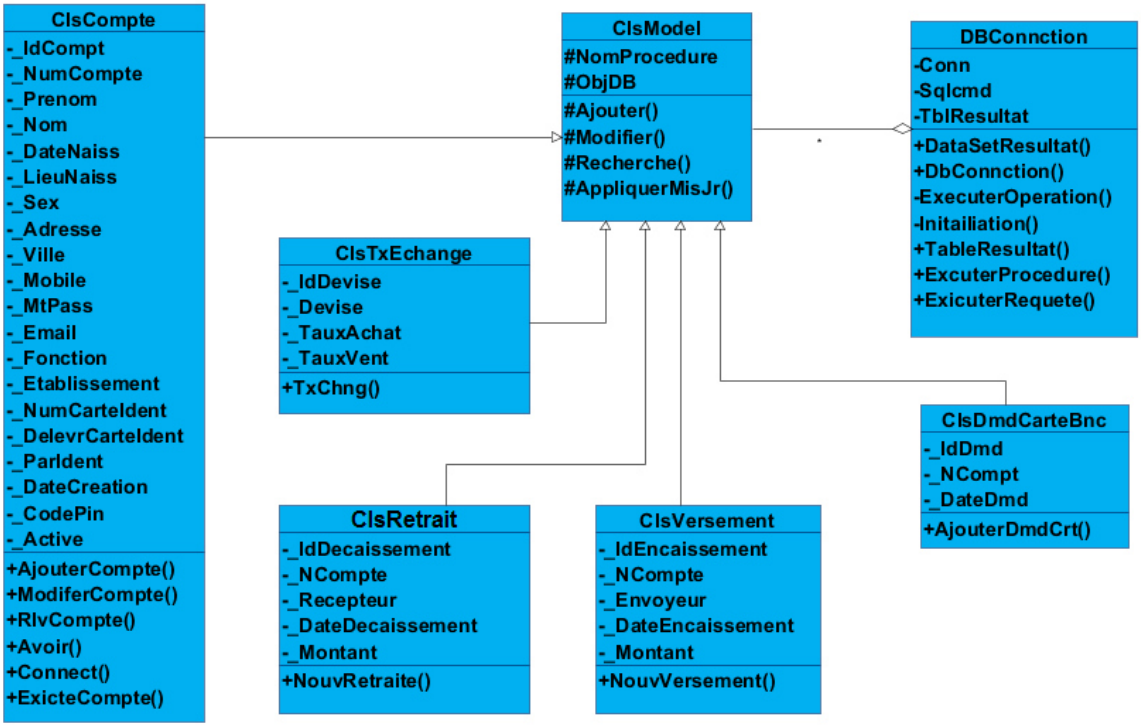


Figure 4-2 : Diagrammedeclasse

- Laclasse(ClsCompte): Cetteclasseestunereprésentationdelacompte bancaireavecseschampsettouteslesopérationsdebase concernantletransaction(AjouterCompte,ModifierCompte...Etc.).
- Laclasse(ClsTxEchange): Cetteclasseestunereprésentationdel'opération de change de devise avec le taux d'achat et de vente .
- Laclasse(ClsRetrait):Cetteclasseestunereprésentationdeladécaissement ou la retrait avecseschamps N° compte,Recepteur,Montantenincluantl'opération(NouvRetrait).

- Laclasse(ClsVersement):Cetteclasseestunereprésentationdel'encaissement ou le versement enincluantl'opération(NouvVersement).
- La classe (ClsModel) : Cette classe donne l'accès a tous autre class appliquer les méthodes ajouter ,modifier, recherche, mise a jour a trévère le nom de procédure.
- La classe (ClsDmdCarteBnc) : Cette classe est une représentation de la demande d'une carte bancaire enincluantl'opération(AjouterDmdCrt).
- Laclasse(DBConnexion):cetteclassecontienttouteslesopérationsliéesà labase dedonnées.

4.1.3 Lediagrammed'Activité(ModélisationDynamique) :

Lecomportementglobaldunotreapplicationserareprésentéeparlediagrammesuivant (figure4-3)

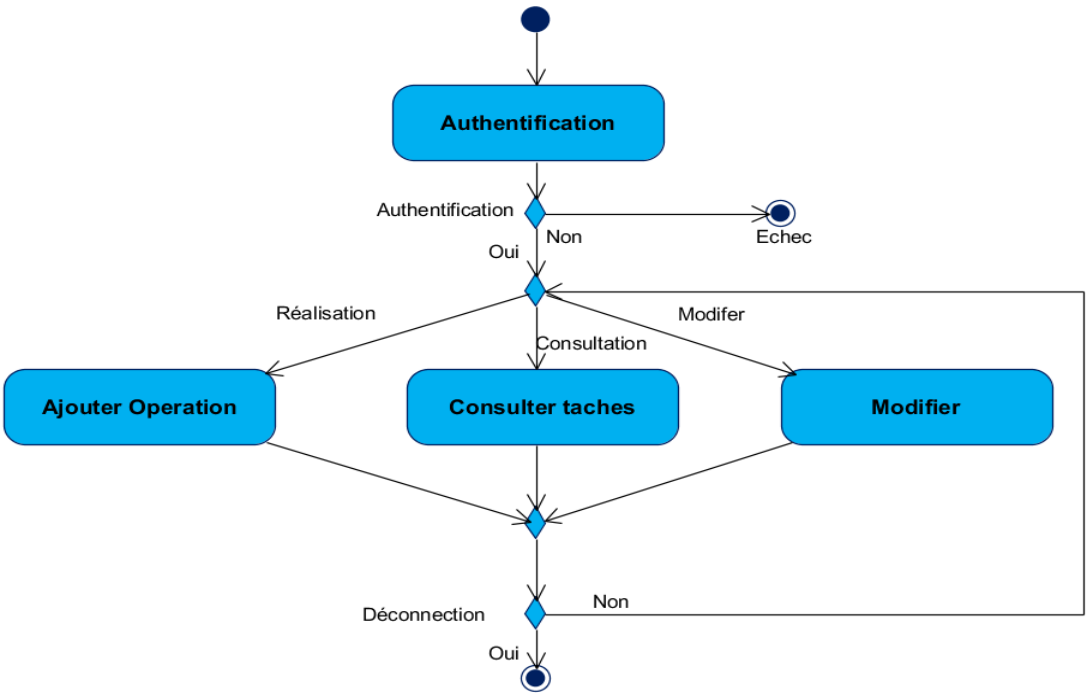


Figure 4-3:*Diagrammed'ActivitéDynamiqueGlobale*

Entrée:numérodecompteetlemotdepasse(Authentification).

Opérations: Ajouterunetransactionouvaliderlestransactionsouvérifierles blocs...etc.

Sortie: lorsqueleclienttermineuneopérationalorsilpeutdéconnectercommeil peutchoisiruneautreopération.

4.1.4 Description desscénarios(ModélisationDynamique) :

Aprèsavoirélaborélesdiagrammesdeclasse,nousallonsreprésenterlesfonctionnaliséesdesFormsetla compositionentreeuxparlesdiagrammesdeséquencecommesuit:

4.1.4.1 Authentification :

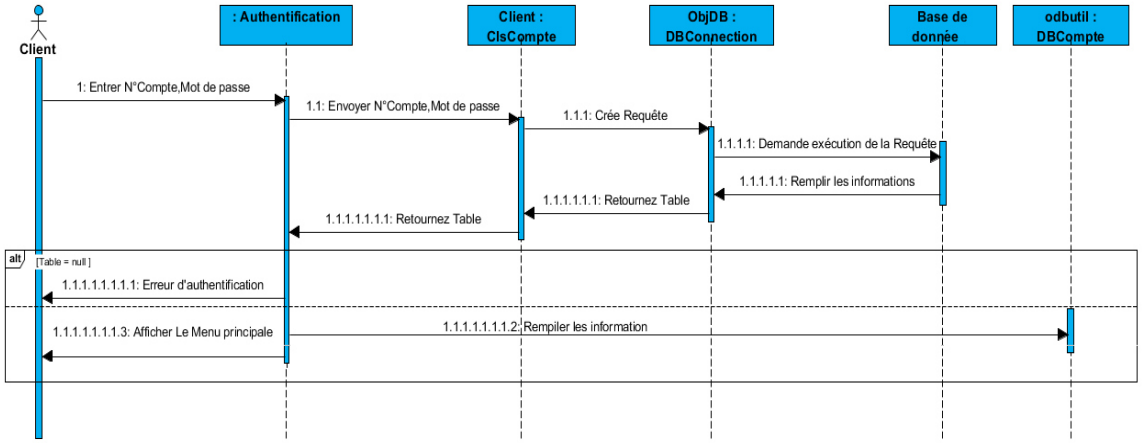


Figure 4-4:Diagrammedeséquenced'Authentification

—*LaForme:Authentification*

1 -Leclientenvoie(leNCompteetleMotdePass),pourfairel'authentification,

2 -LaclasseClsComptevérifies'ilpeutconnecterounonparlacréationd'unerequête,

3 -

Cetterequêteestexécutéeparl'objetDBConnectionquivainterrogerlabasededonnéespourextrair el'information,

-Silerésultatdel'exécutionderequêteestnégative,celasignifiequ'ilyaune erreurd'Authentification,

-SinonlaformeAuthentificationremplielesinformationsetentreraumenuprincipal.

4.1.4.2 NouveauClient :

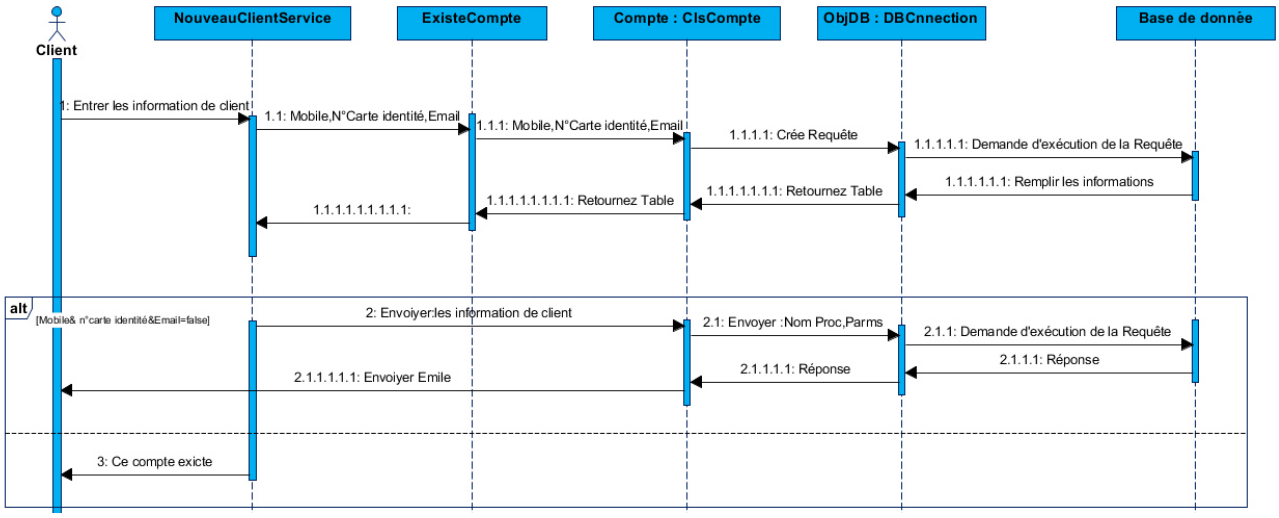


Figure 4-5:*DiagrammedeséquenceNouvelle*Client

—LaForme:Nouveau Client :

-pourajouter un nouveau clientil faut d'abord vérifier qu'il n'existe pas,

Par la saisirde leurinformations tell que numéro mobile,N°carte identité et email (qui il sont des champs unique),la requête envoyé aux base de donnes.

-LaclasseClsComptevérifielapossibilitédelaconnectionparlacréationd'unerequête,

- Cetterequêtevaêtreexécutéeparl'objetDBConnectionquivainterrogerla basededonnéespourextrairel'information,

-LerésultatestretournéversLaformeComptepourrépondreauclientpar l'envoied'unmessage.

4.1.4.3 Avoir :

Selon le numéro de compte en peut consulter le solde.

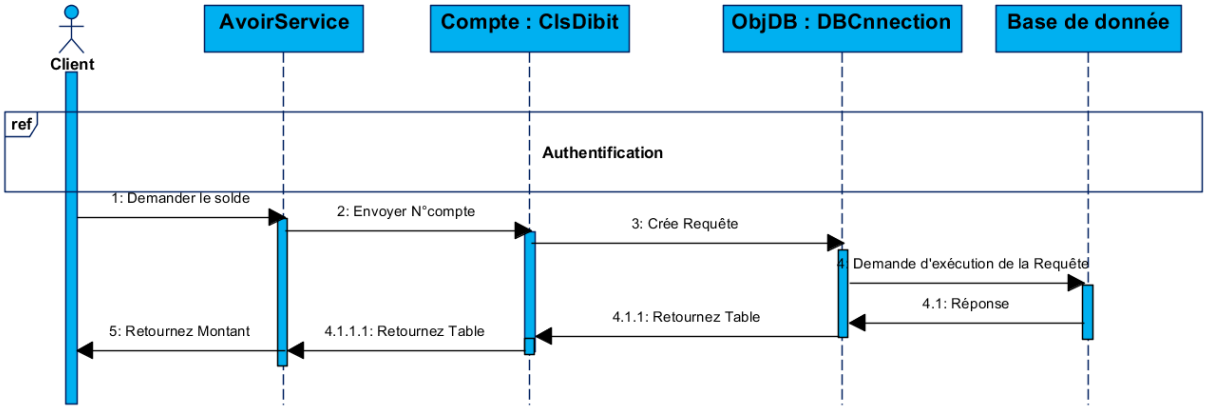


Figure 4-6:Diagrammedeséquenced'avoir

4.1.4.4 Versement :

Pour effectué un nouveau versement il faut entrer le numéro de compte client et le montant averser.

L'opération de versement suivre plusieurs actions et test pour retournétrue (compte existe et l'opération bien effectue)ou false(le compte n'existe pas) .

Le service Existe Compte commence la procédure de recherche selon le numéro compte(en peut ajouter les champs numéro carte identité ; email, numéro téléphone qu'ont des champ unique pour chacun par défaut sont déclarer nul)

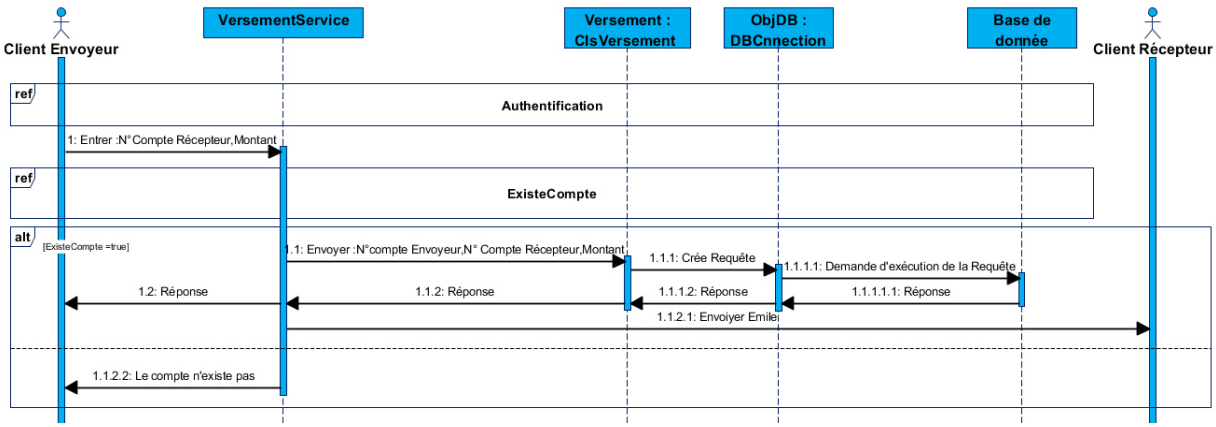


Figure 4-7 :Diagramme de séquencedeversement

4.1.4.5 Retrait :

L'opération du retrait similaire à opération de versement, sauf qu'on débite le compte.

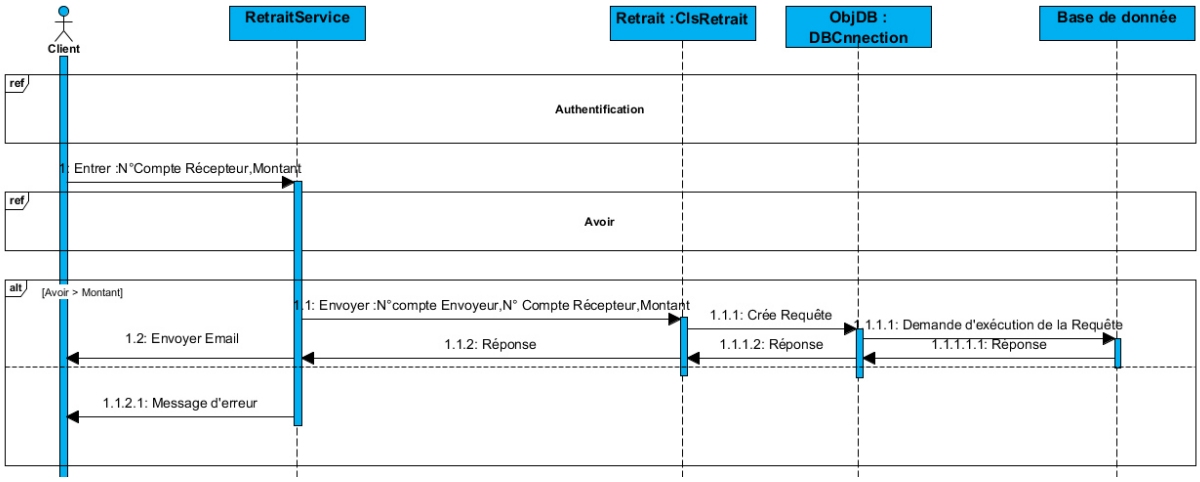


Figure 4-8:Diagrammedeséquencede retrait

4.1.4.6 Transfert :

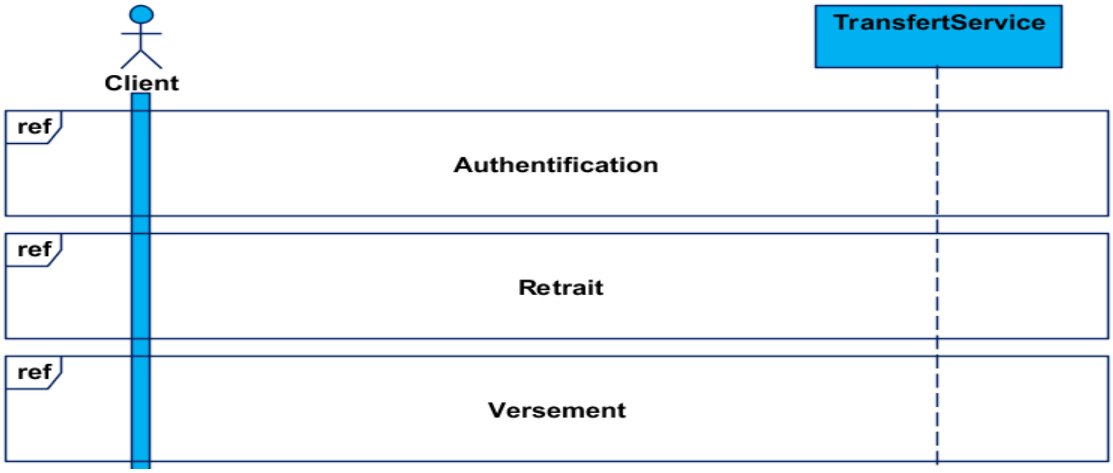


Figure 4-9:*Diagrammedeséquencede transert*

—LaForme:AfficherLesBlocs

-Aprèsl'Authentification,leclientdemanded'afficheretenvoielesinformations.

-Cetterequêteestexécutéeparl'objetDBConnectionquivainterrogerlabase

dedonnéespourextrairel'information.

-cetterequêteestexécutéeparl'objetDBConnectionquivainterrogerlabasede donnéespourextraire.

4.2- Lasécuritédel'application

Laplupartdesapplicationssontunecibleprivilègedesattaquescarellessontfacilement accessiblesetoffrentunpointd'entréelucratif,pouraccéderà desdonnéesprécieuses.

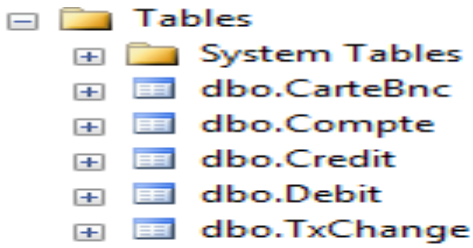
Cesmenacespourraientconduireà desrésultatscatastrophiquesenparticulierdansledomainedesbanques.

Pourluttercontrelesattaquescomplexesetdistribuées,lesentreprisesontbesoindeprotégerleursapplications contredesmenacesnouvelleset émergentes,sansaffecterLesperformancesouladisponibilitédesapplications.Pourcelanousavonsutiliépournotre applicationdifférentsniveauxdeprotectionquisont:

4.2.1 Labasededonnées :

Apartirdenotre étudefaiteetlesdiagrammesUML établis,nousavonsréaliséune basededonnéesrelationnellequicontientdestablespouraccéderauxbasesdedonnées:

Lestables :



4.3- L'environnementsoftwaredesystème :

4.3.1 Laplateforme.NETFramework :

Le.NETFrameworkestuneplateformededéveloppementlargementutiliséepourla créationd'applicationsdestinéesàWindows,WindowsStore,WindowsPhone,Windows Server et Windows Azure. La plateforme .NET Framework comprend les langages de programmation JavaScript et Visual Basic, le Common Language Runtime, ainsi qu'une abondantebibliothèquedeclasses.

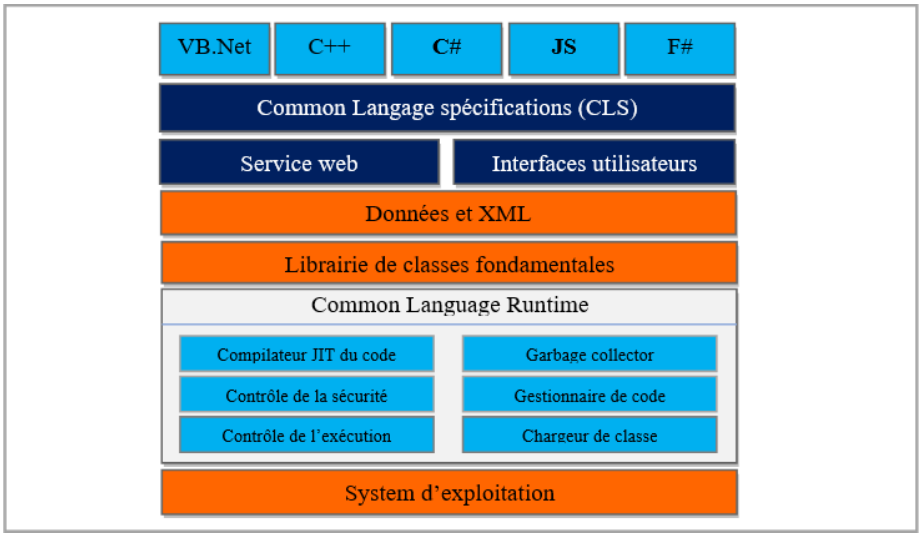


Figure 4-10 : Architecteur.NetFramework[26]

Nousavonsadaptépourcettetechnologiepourlesraisonssuivantes:

Fournirunenvironnementcohérentdeprogrammationorientéeobjetquelecodeobjetsoitstockéetexécutélocalement,exécutélocalementmaisdistribuésurInternet ouexécutéà distance.

Fournirunenvironnementd'exécutiondecodequiminimiseledéploiementdelogicielsetdeconflitsdeversions.

Fournirunenvironnementd'exécutiondecodequigarantitl'exécutionsécuriséede code

ycomprislecodecrééparuntiersd'unniveaudeconfiancemoyenouuntiers inconnu.

Générertouteslescommunicationsàpartirdesnormesd'industriespours'assurer quelecodebasésurleFramework.NETpeuts'intégreràn'importequelautrecode.

4.3.2 LeLangageC# :

Le langage star de la nouvelle version de Visual Studio et de l'architecture. NET est C#,unlangagedérivéduC++.Ilreprendcertainescaractéristiquesdeslangagesapparusces dernièresannéesetenparticulierdeJava(quireprenaitdéjàà soncomptedesconcepts introduitsparSmalltalkquinzeansplustôt)maistrèsrapidement,C#ainnovéetles conceptsainsiintroduitssontaujourd'huicommunémentreprisdanslesautreslangages. C#peutêtreutilisépourcréer,avecunefacilitéincomparable,desapplicationsWindows etWeb.C#devientlelangagedeprédilectiond'ASP.NETquipermetdecréerdespages Webdynamiquesavecprogrammationcôtéserveur[21].

4.3.3 SQLServer :

SQLServerestunsystèmedegestiondebasesdedonnéesrelationnelles(SGBDR) répondantauxexigencesprofessionnellesdustockagededonnées.SQLServerprenden chargenativementpourlacommunicationderequêtesentreclientetserveur:

SQLServerintègrepardéfautdesoutilsdegestion,d'administrationetdedévelop- pementdebasesdedonnées.

Déploiementparunsetup,miseenœuvreetadministrationpardesinterfacesgra- phiquesintuitives.

Gestionavancéedelasécuritéenoffrantdeuxmodesd'authentification(Authenti- ficationWindowsetAuthentificationSQLServer).

CoûtrelativementmoinscherparrapportauxautresSGBDdumarché[28].

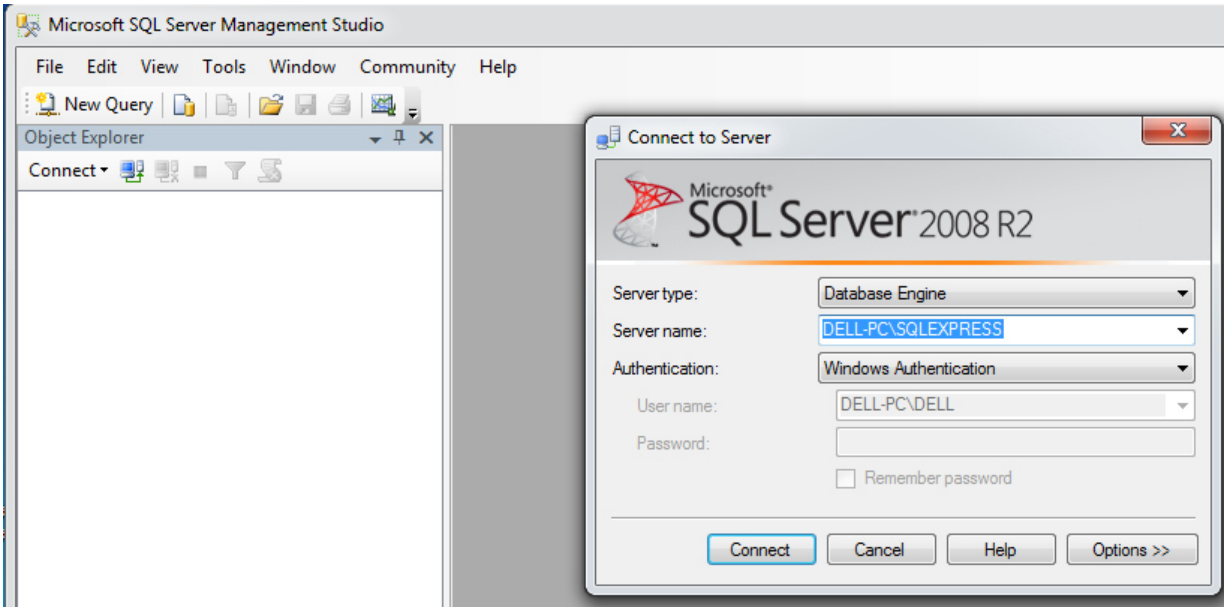


Figure 4-11: *SQLServer2008*

4.3.4 ASP.NET(ActiveServerPage) :

ASP.NETestunensembledetechnologiesdeprogrammationwebcréépar Microsoft. Lesprogrammeurspeuventutiliser ASP.NET pourcréerdessiteswebdynamiques,desapplicationsweboudeswebservicesXML.Latechnologieestaccessiblegrâceà l'installation d'unserveurwebcompatibleASP(IIS)ouà l'intérieurdeVisualWebDéveloppeur Express Edition. ASP.NET fait partie de la plateforme Microsoft .NET et est le successeur de latechnologie Active Server Pages (ASP) [27], nous avons apte pour cette technologie pourlesraisonssuivantes:

ASP.NETréduitconsidérablementlaquantitédecodenécessairepourconstruirede grandesapplications.

Touslesprocessussontétroitementsurveillésetgérésparleruntime WindowsForms, desortequesileprocessusestmort,unnouveauprocessuspeutêtrecréé àsaplace,

cequiaideà gardervotreapplicationdisponibleenpermanencepourtraiter les demandes.

Lesfonctionnefacilementavec ADO.NETenutilisantlaliaisondedonnéesetdesfonctionnalitésmiseenpage.Ilestuneapplicationquifonctionneplusrapidementetde comptoirsdegrandsvolumesd'utilisateurssansavoirdesproblèmesdeperformance [27].

4.3.5 ADO.NET(ActiveXDataObjects)

ADO.NETestunensembledeclassesquiexposentlesservicesd'accèsauxdonnéespour les programmeurs .NET Framework. ADO.NET propose un large ensemble de composantspourlacréationd'applicationsdistribuéesavecpartagededonnées.Partieintégrantedu

.NETFramework,ilpermetd'accéderàdesdonnéesrelationnelles,XMLetd'application. ADO.NETrépondà diversbesoinsenmatièrededéveloppement,enpermettantnotamment decréerdesclientsdebasesdedonnéesfrontauxetdesobjetsmétierdecouche intermédiaireutilisépardesapplications,outils,langagesounavigateursInternet[27],on aaptepourcettetechnologiepourlesraisonssuivantes:

- Interopérabilité
- Facilitédemaintenance
- Facilitédeprogrammation
- Performances
- Evolutivité

4.3.6 ServeurIIS(InternetInformationServer) :

IISestleserveurWebdeMicrosoft.IlcontrôlelespagesWeb(celles évidemmentqui sonthébergéessursamachine)etenvoieleHTMLdecespagesWebauxnavigateursdes clientsquienfontlademande[27].

4.3.7 VisualstudioIDE(IntegratedDevelopmentEnvironment) :

VisualStudioestunensemblecompletd'outilsdedéveloppementpermettantdegénérerdes applications Web ASP.NET, des Services Web XML, des applications bureautiques etdes applications mobiles. Visual Basic, Visual C# et Visual javascript utilisent tous lemêmeenvironnementdedéveloppementintégré(IDE),quipermetlepartaged'outilset facilitelacréationdesolutionsàplusieurslangages.Parailleurs,ceslangagesutilisentles fonctionnalitésdu.NETFramework,quifournitunaccèsàdestechnologiescléssimplifiant ledéveloppementd'applicationsWebASPetdeServicesWebXML[27],lesavantages suivantsVisualStudioIDE:Detrèsnombreuxoutilssontdisponiblesetpeuventinteragir.

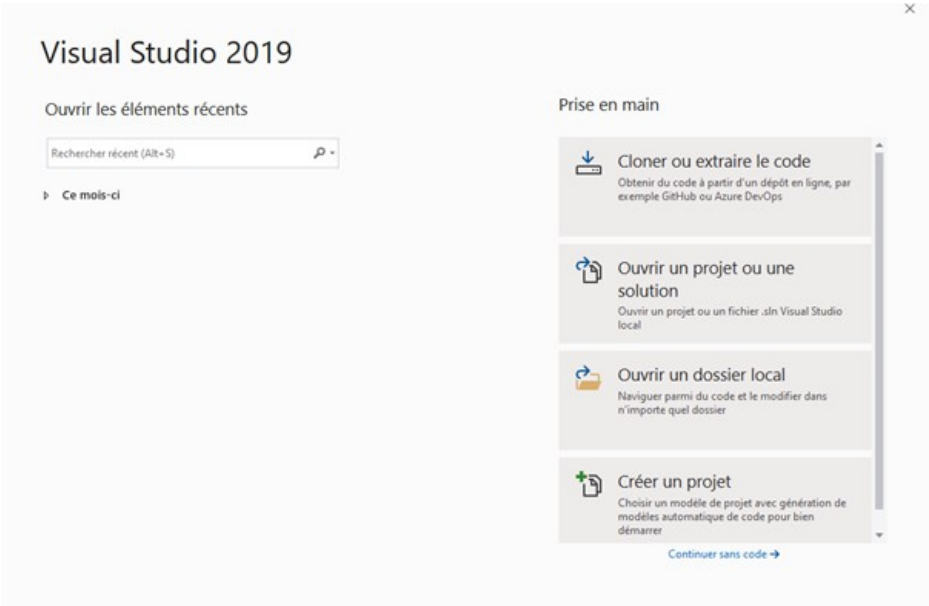


Figure 4-12 : ***VisualstudioCommunity2019***

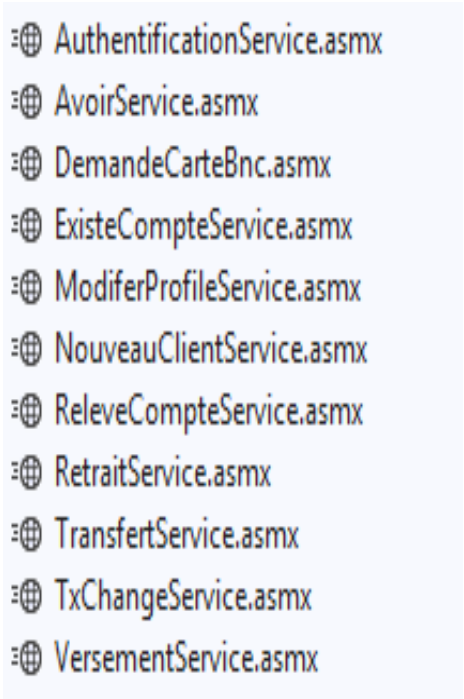


Figure 4-13 : Les services existant

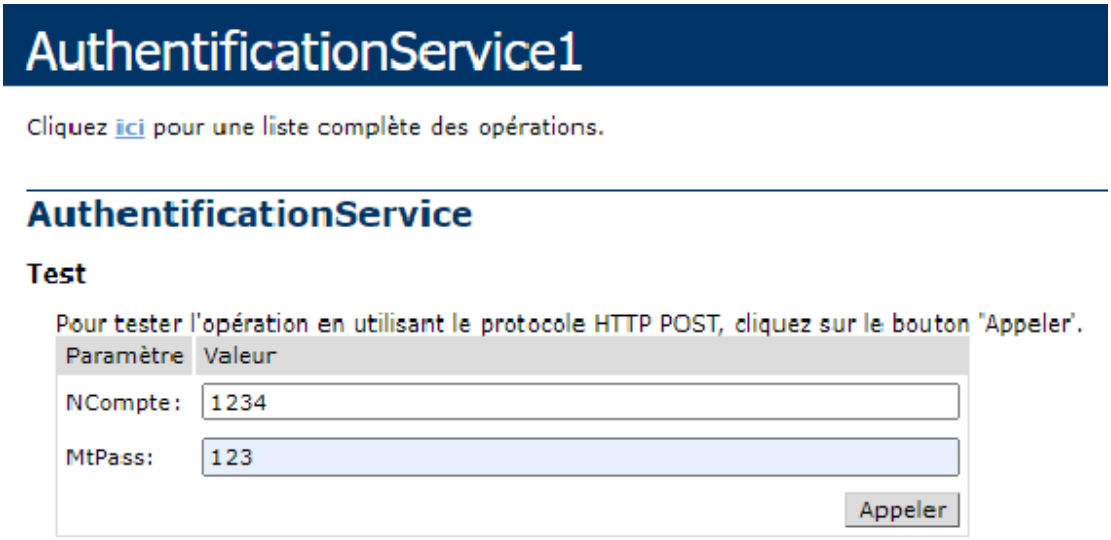


Figure 4-14 : Services Authentification

```
<?xml version="1.0" encoding="UTF-8"?>
<boolean xmlns="http://tempuri.org/">true</boolean>
```

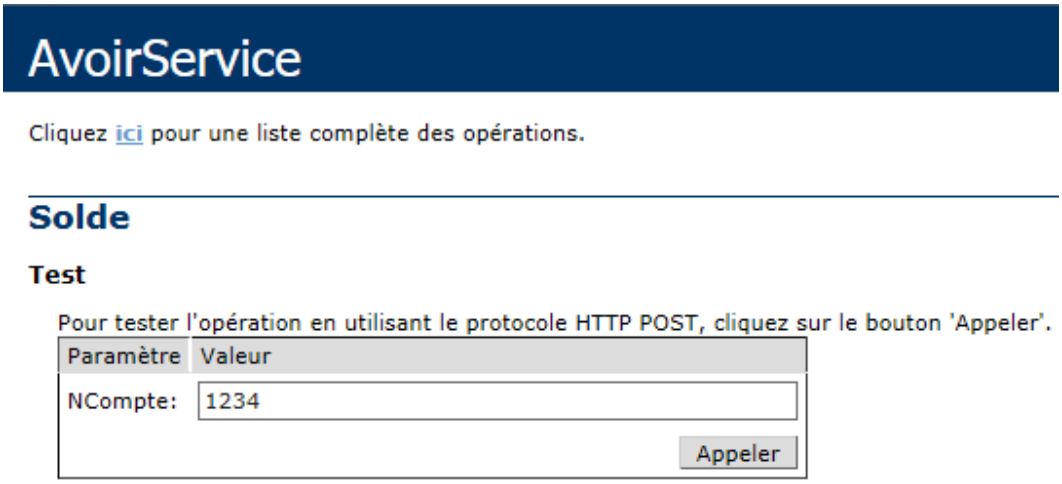


Figure 4-15 : Service Avoir

Pour savoir le solde il suffit d'entrer le numéro de compte d'un client.

```
<?xml version="1.0" encoding="UTF-8"?>
<doublexmlns="http://tempuri.org/">11900,0000</double>
```

SOAP 1.1

Le texte suivant est un exemple de demande et de réponse SOAP 1.1. Les espaces réservés affichés doivent être remplacés par des valeurs réelles.

```
POST /Services/AvoirService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/Solde"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Solde xmlns="http://tempuri.org/">
      <NCompte>string</NCompte>
    </Solde>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SoldeResponse xmlns="http://tempuri.org/">
      <SoldeResult>string</SoldeResult>
    </SoldeResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP 1.1

Le texte suivant est un exemple de demande et de réponse SOAP 1.1. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

POST /Services/AvoirService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: **length**
SOAPAction: "http://tempuri.org/Solde"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelopexmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<Soldexmlns="http://tempuri.org/">
<NCompte>**string**</NCompte>
</Solde>
</soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: **length**

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelopexmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<SoldeResponsexmlns="http://tempuri.org/">
<SoldeResult>**string**</SoldeResult>
</SoldeResponse>
</soap:Body>
</soap:Envelope>

Figure 4-16 : services demande carte bancaire

SOAP 1.1

Le texte suivant est un exemple de demande et de réponse SOAP 1.1. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST /Services/DemandeCarteBnc.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/DomandeCarteBancaire"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelopexmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<DomandeCarteBancairexmlns="http://tempuri.org/">
<NCompte>string</NCompte>
</DomandeCarteBancaire>
</soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelopexmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<DomandeCarteBancaireResponsexmlns="http://tempuri.org/">
<DomandeCarteBancaireResult>string</DomandeCarteBancaireResult>
</DomandeCarteBancaireResponse>
</soap:Body>
</soap:Envelope>
```

La réponse à la demande d'une carte bancaire :

```
<?xml version="1.0" encoding="UTF-8"?>
<string xmlns="http://tempuri.org/">Le Processus a été Ajouter Avec Succès </string>
```

Figure 4-17 : Services Existe Compte

```
<?xml version="1.0" encoding="UTF-8"?>
<booleanxmlns="http://tempuri.org/">true</boolean>
```

Si le compte n'existe pas, la réponse retourné false

```
<?xml version="1.0" encoding="UTF-8"?>
<boolean xmlns="http://tempuri.org/">false</boolean>
```

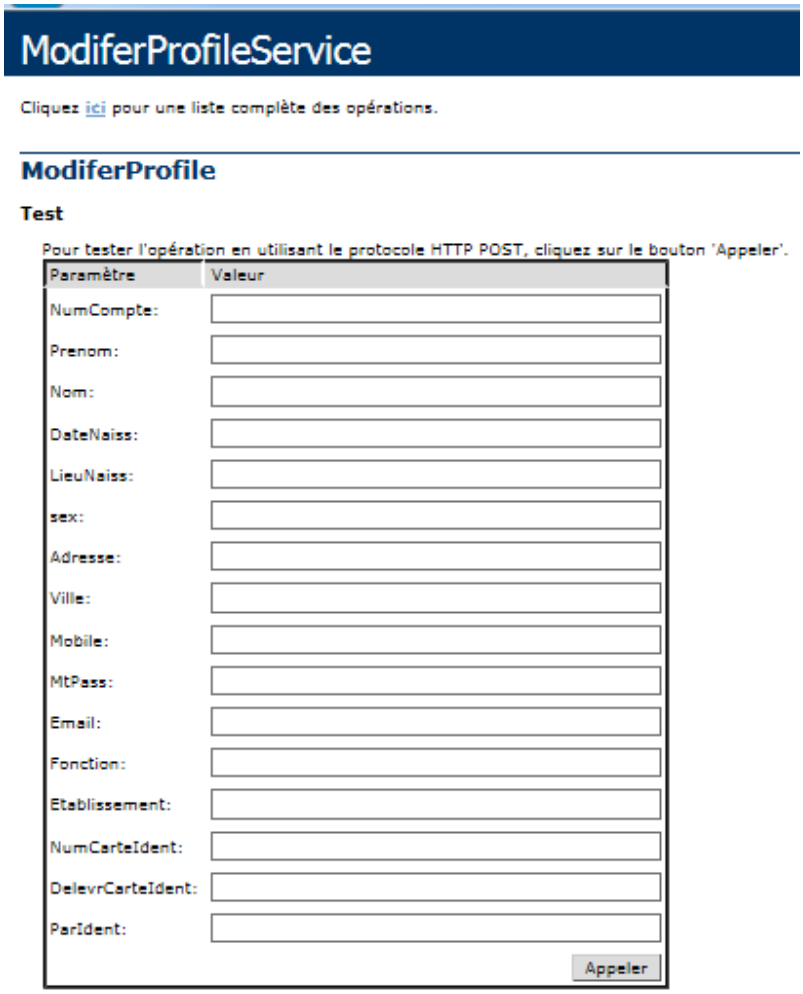


Figure 4-18 : Modifier profil d'un client

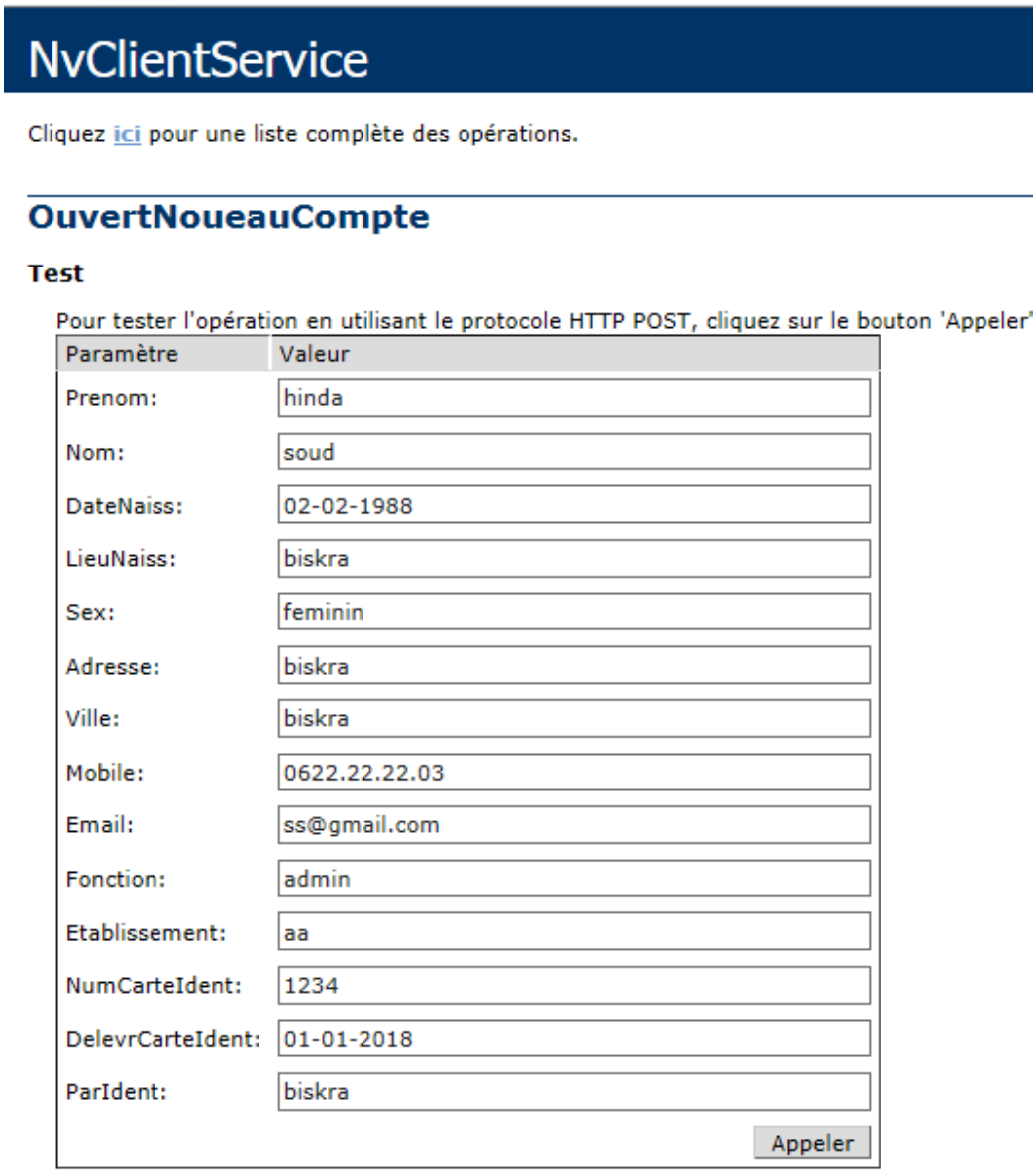


Figure 4-19 : Service ouvrir un nouveau compte

```
<?xml version="1.0" encoding="UTF-8"?>
<string xmlns="http://tempuri.org/">Le Processus a été Ajouter Avec Succès </string>
```

SOAP 1.1

Le texte suivant est un exemple de demande et de réponse SOAP 1.1. Les **espaces réservés** affichés doivent être remplacés par des valeurs réelles.

```
POST /Services/NouveauClientService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/OuvertNoueauCompte"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelopexmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<OuvertNoueauComptexmlns="http://tempuri.org/">
<Prenom>string</Prenom>
<Nom>string</Nom>
<DateNaiss>dateTime</DateNaiss>
<LieuNaiss>string</LieuNaiss>
<Sex>string</Sex>
<Adresse>string</Adresse>
<Ville>string</Ville>
<Mobile>string</Mobile>
<Email>string</Email>
<Fonction>string</Fonction>
<Etablissement>string</Etablissement>
<NumCarteIdent>string</NumCarteIdent>
<DelevrCarteIdent>dateTime</DelevrCarteIdent>
<ParIdent>string</ParIdent>
</OuvertNoueauCompte>
</soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelopexmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<OuvertNoueauCompteResponsexmlns="http://tempuri.org/">
<OuvertNoueauCompteResult>string</OuvertNoueauCompteResult>
</OuvertNoueauCompteResponse>
</soap:Body>
</soap:Envelope>
```

DELL-PC\SQLEXPR...que - dbo.Compte | DELL-PC\SQLEXPRE...nque - dbo.Credit

IdCompte	NCompte	Nom	Prenom	DateNaiss	LieuNiaiss	Sex	Adresse	Ville	Mobile	MtPass
1	1234	SOUDADI	BAHIA	1985-02-27	BISKRA	FEMININ	BISKRA	BISKRA	0778232556	123
2	12345	ali	mohamed	2001-01-01	biskra	masculin	biskra	biskra	0666666666	456
3	bna123	soud	hinda	1988-02-02	biskra	feminin	biskra	biskra	0622.22.22.03	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tableau 4-1 : La table de base de données compte

Figure 4-20 : Service relever de compte

Figure 4-21 : Service retrait

SOAP 1.2

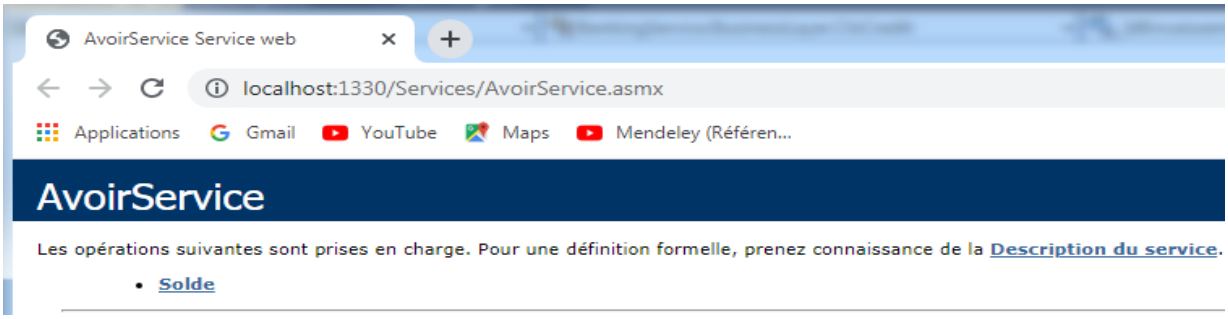
Le texte suivant est un exemple de demande et de réponse SOAP 1.2. Les espaces réservés affichés doivent être remplacés par des valeurs réelles.

```
POST /Services/RetraiteService.asmx HTTP/1.1
Host: localhost
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <Retraite xmlns="http://tempuri.org/">
      <NCompte>string</NCompte>
      <Recepteur>string</Recepteur>
      <Montant>double</Montant>
    </Retraite>
  </soap12:Body>
</soap12:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <RetraiteResponse xmlns="http://tempuri.org/">
      <RetraiteResult>boolean</RetraiteResult>
    </RetraiteResponse>
  </soap12:Body>
</soap12:Envelope>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<boolean xmlns="http://tempuri.org/">true</boolean>
```

9	1234	12345	2021-06-28 20:40:...	2000...



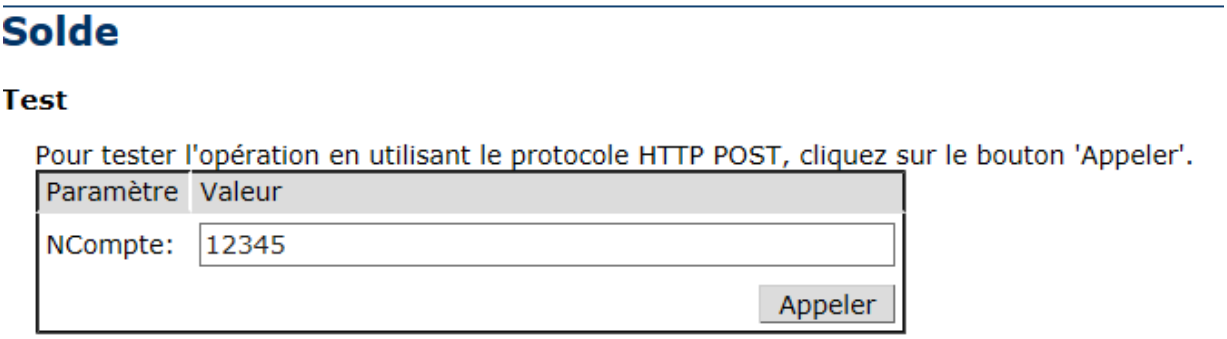


Figure 4-22 : Service solde

```
<?xml version="1.0" encoding="UTF-8"?>
<doublexmlns="http://tempuri.org/">14750,0000</double>
```

Cas erreur d'entrée :

System.ArgumentException: Impossible de convertir fghhj en System.Double.
Nom du paramètre : type --->System.FormatException: Le format de la chaîne d'entrée est incorrect.
à System.Number.ParseDouble(String value, NumberStyles options, NumberFormatInfonumfmt)
à System.String.System.IConvertible.ToDouble(IFormatProvider provider)
à System.Convert.ChangeType(Object value, Type conversionType, IFormatProvider provider)
à System.Web.Services.Protocols.ScalarFormatter.FromString(String value, Type type)
--- Fin de la trace de la pile d'exception interne ---
à System.Web.Services.Protocols.ScalarFormatter.FromString(String value, Type type)
à System.Web.Services.Protocols.ValueCollectionParameterReader.Read
(NameValueCollection collection)
à System.Web.Services.Protocols.HtmlFormParameterReader.Read(HttpRequest request)
à System.Web.Services.Protocols.HttpServerProtocol.ReadParameters()
à System.Web.Services.Protocols.WebServiceHandler.CoreProcessRequest()

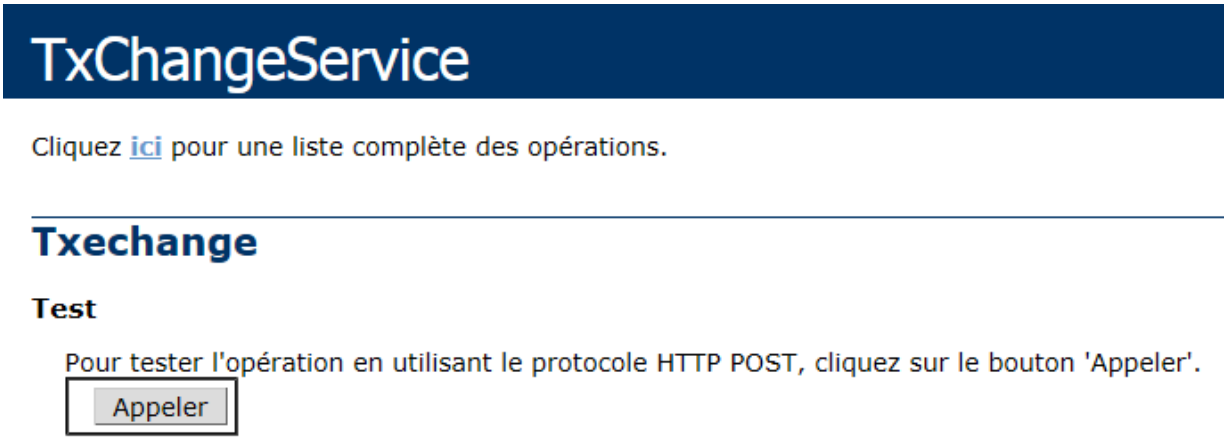


Figure 4-23 : Service taux de change

```
<?xml version="1.0" encoding="UTF-8"?>
- <ArrayOfTauxChange xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    - <TauxChange>
        <IdDevise>1</IdDevise>
        <Devise>100</Devise>
        <TauxAchat/>
        <TauxVent/>
      </TauxChange>
    - <TauxChange>
        <IdDevise>2</IdDevise>
        <Devise>200</Devise>
        <TauxAchat/>
        <TauxVent/>
      </TauxChange>
    - <TauxChange>
        <IdDevise>3</IdDevise>
        <Devise>300</Devise>
        <TauxAchat/>
        <TauxVent/>
      </TauxChange>
  </ArrayOfTauxChange>
```

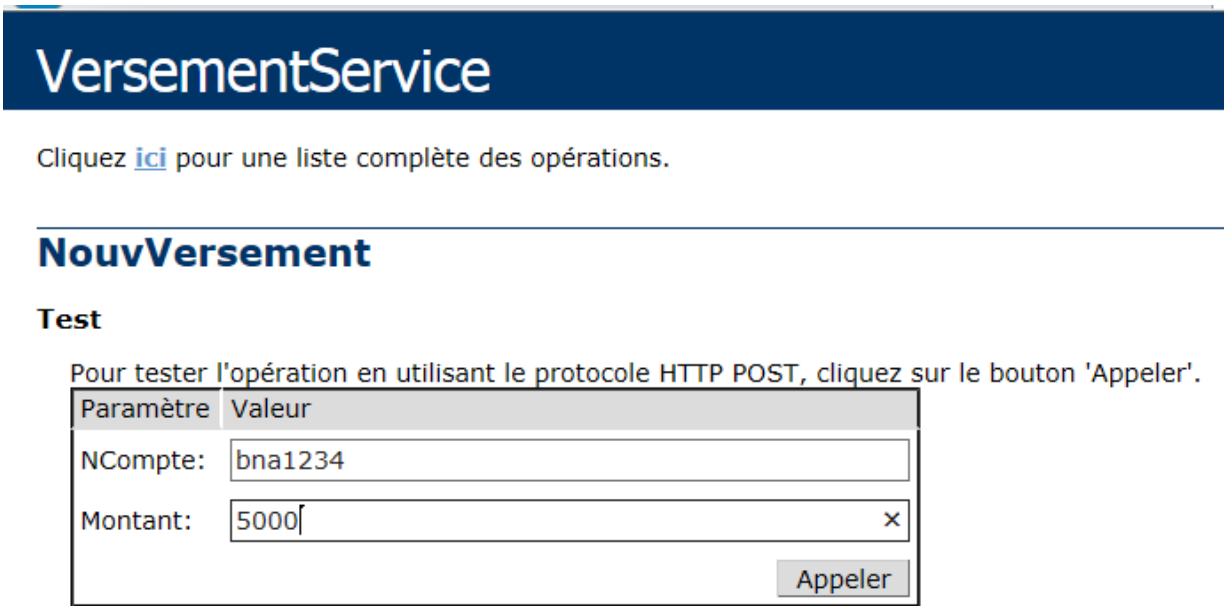


Figure 4-24 : Service versement exemple 1

Cas erreur de saisir le numéro de compte ; la réponse est false

```
<?xml version="1.0" encoding="UTF-8"?>
<boolean xmlns="http://tempuri.org/">false</boolean>
```

Figure 4-25 : Service versementexemple 2

```
<?xml version="1.0" encoding="UTF-8"?>
<boolean xmlns="http://tempuri.org/">true</boolean>
```

Fichier XML d'un service versement :

```
<?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://tempuri.org/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://tempuri.org/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/">
  - <wsdl:types>
    - <s:schema targetNamespace="http://tempuri.org/" elementFormDefault="qualified">
      - <s:element name="NouvVersement">
        - <s:complexType>
          - <s:sequence>
              <s:element name="NCompte" type="s:string" maxOccurs="1" minOccurs="0"/>
              <s:element name="Montant" type="s:double" maxOccurs="1" minOccurs="1"/>
            </s:sequence>
          </s:complexType>
        </s:element>
      - <s:element name="NouvVersementResponse">
        - <s:complexType>
          + <s:sequence>
          </s:complexType>
        </s:element>
      </s:schema>
    </wsdl:types>
  - <wsdl:message name="NouvVersementSoapIn">
      <wsdl:part name="parameters" element="tns:NouvVersement"/>
    </wsdl:message>
  - <wsdl:message name="NouvVersementSoapOut">
      <wsdl:part name="parameters" element="tns:NouvVersementResponse"/>
    </wsdl:message>
  - <wsdl:portType name="VersementServiceSoap">
    - <wsdl:operation name="NouvVersement">
        <wsdl:input message="tns:NouvVersementSoapIn"/>
        <wsdl:output message="tns:NouvVersementSoapOut"/>
      </wsdl:operation>
    </wsdl:portType>
  - <wsdl:binding name="VersementServiceSoap" type="tns:VersementServiceSoap">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    - <wsdl:operation name="NouvVersement">
        <soap:operation style="document" soapAction="http://tempuri.org/NouvVersement"/>
      - <wsdl:input>
          <soap:body use="literal"/>
        </wsdl:input>
      - <wsdl:output>
          <soap:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  - <wsdl:binding name="VersementServiceSoap12" type="tns:VersementServiceSoap">
      <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    - <wsdl:operation name="NouvVersement">
        <soap12:operation style="document" soapAction="http://tempuri.org/NouvVersement"/>
      - <wsdl:input>
          <soap12:body use="literal"/>
        </wsdl:input>
      - <wsdl:output>
          <soap12:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  - <wsdl:service name="VersementService">
    - <wsdl:port name="VersementServiceSoap" binding="tns:VersementServiceSoap">
        <soap:address location="http://localhost:1330/Services/VersementService.asmx"/>
      </wsdl:port>
    - <wsdl:port name="VersementServiceSoap12" binding="tns:VersementServiceSoap12">
        <soap12:address location="http://localhost:1330/Services/VersementService.asmx"/>
      </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

4.4- Principe de perturbation :

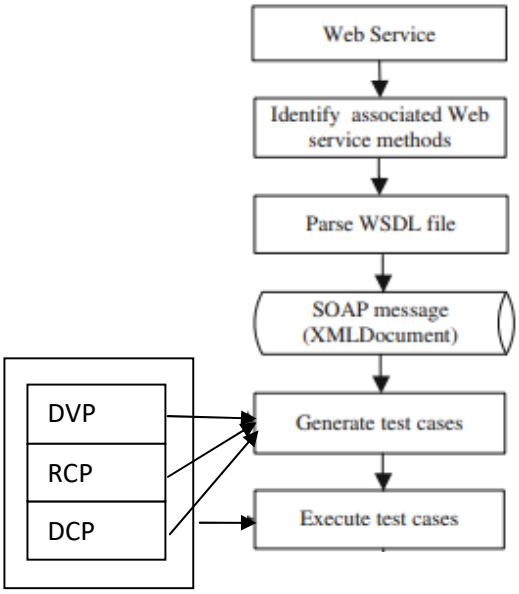


Figure 4-26:L'organigramme du système de test des services Web

Perturbation de la valeur des données (DVP) : qui est effectuée en modifiant les valeurs dans un message SOAP.

Exemple de la perturbation de valeur de données à la méthode nouveau versement ()

Valeurd'origine	Valeursperturbées	Cas de test
<NuméroCompte>BNA1234</NuméroCompte>	00000000000000000000000000	maximumlength
	—	minimumlength
<Montant>2000.25</Montant>	$2^{63}-1$	maximumvalue
	-2^{63}	minimumvalue
	0	zero

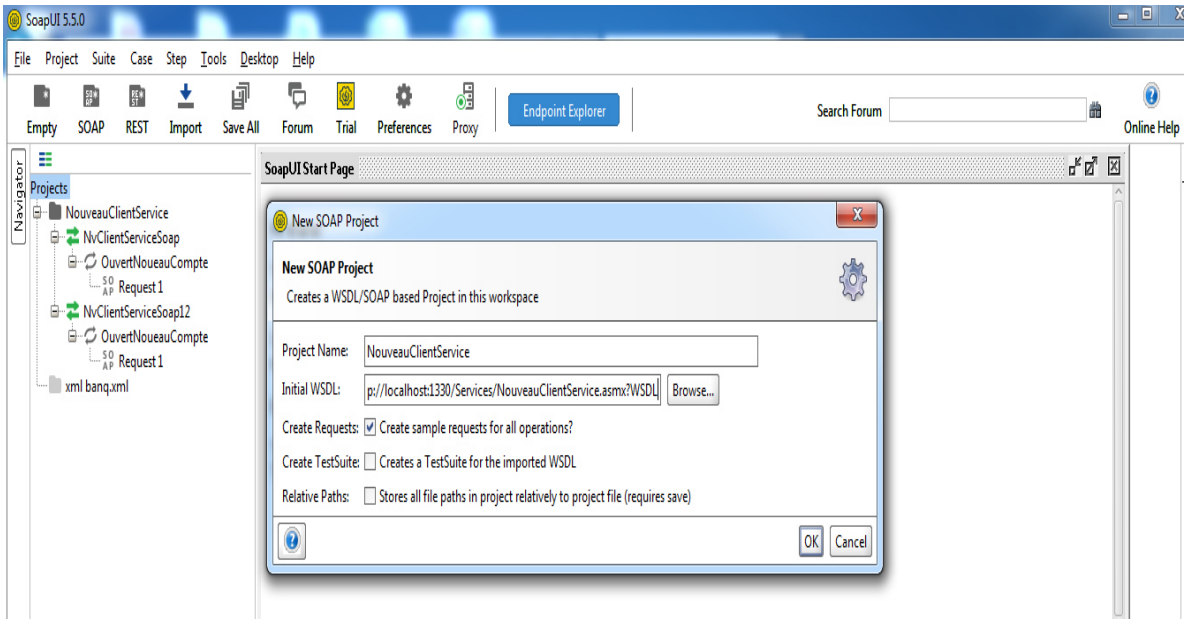
Tableau 4-2:Example 1 de DVP

exemple perturbation de la valeur de données au méthode ouvrir nouveau Compte()

Valeurd'origine	Valeursperturbées	Cas de test
<Prenom>**bahia**</Prenom>	00000000000000000000000000	maximumlength
	—	minimumlength
<Nom>**soudadi**</Nom>	00000000000000000000000000	maximumlength
	—	minimumlength
<DateNaiss>**dateTime**</DateNaiss>	$2^{32}-1$	maximumvalue
	-2^{32}	minimumvalue
	0	Zero
<LieuNaiss>**biskra**</LieuNaiss>	00000000000000000000000000	maximumlength
	—	minimumlength
<Sex>**feminin**</Sex>	00000000000000000000000000	maximumlength
	—	minimumlength
<Adresse>**biskra**</Adresse>	00000000000000000000000000	maximumlength
	—	minimumlength
<Ville>**biskra**</Ville>	00000000000000000000000000	maximumlength
	—	minimumlength
<Mobile>**0778-23-22-22**</Mobile>	00000000000000000000000000	maximumlength
	—	minimumlength
<Email>**baho@gmailcom**</Email>	00000000000000000000000000	maximumlength
	—	minimumlength
<Fonction>**Ing**</Fonction>	00000000000000000000000000	maximumlength
	—	minimumlength
<Etablissement>**aa**</Etablissement>	00000000000000000000000000	maximumlength
	—	minimumlength
<NumCarteIdent>**35791**</NumCarteIdent>	00000000000000000000000000	maximumlength
	—	minimumlength
<DelevrCarteIdent>**01-01-2018**</DelevrCarteIdent>	$2^{32}-1$	maximumvalue
	-2^{32}	minimumvalue
	0	Zero
<ParIdent>**biskra**</ParIdent>	00000000000000000000000000	maximumlength
	—	minimumlength

Tableau 4-3: Exemple 2 de DVP

Perturbation de la communication RPC :



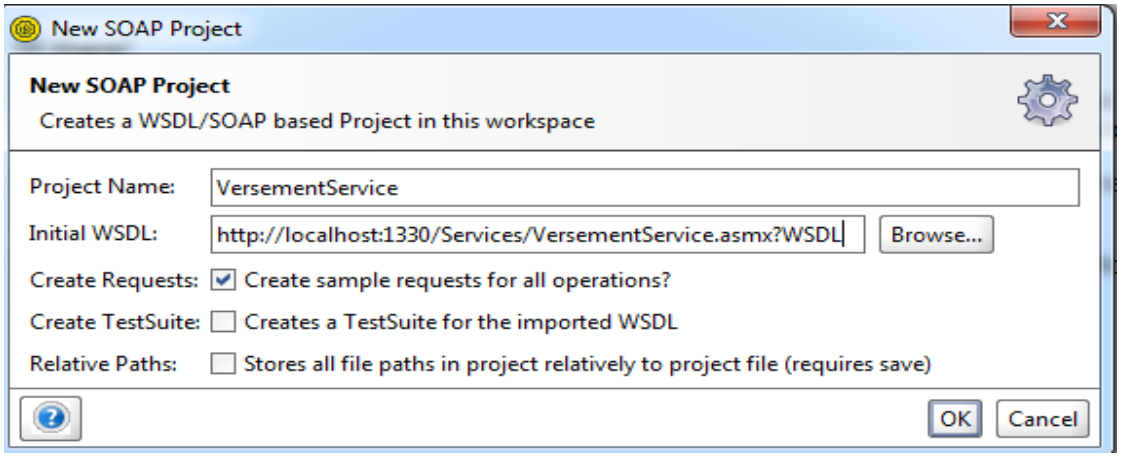
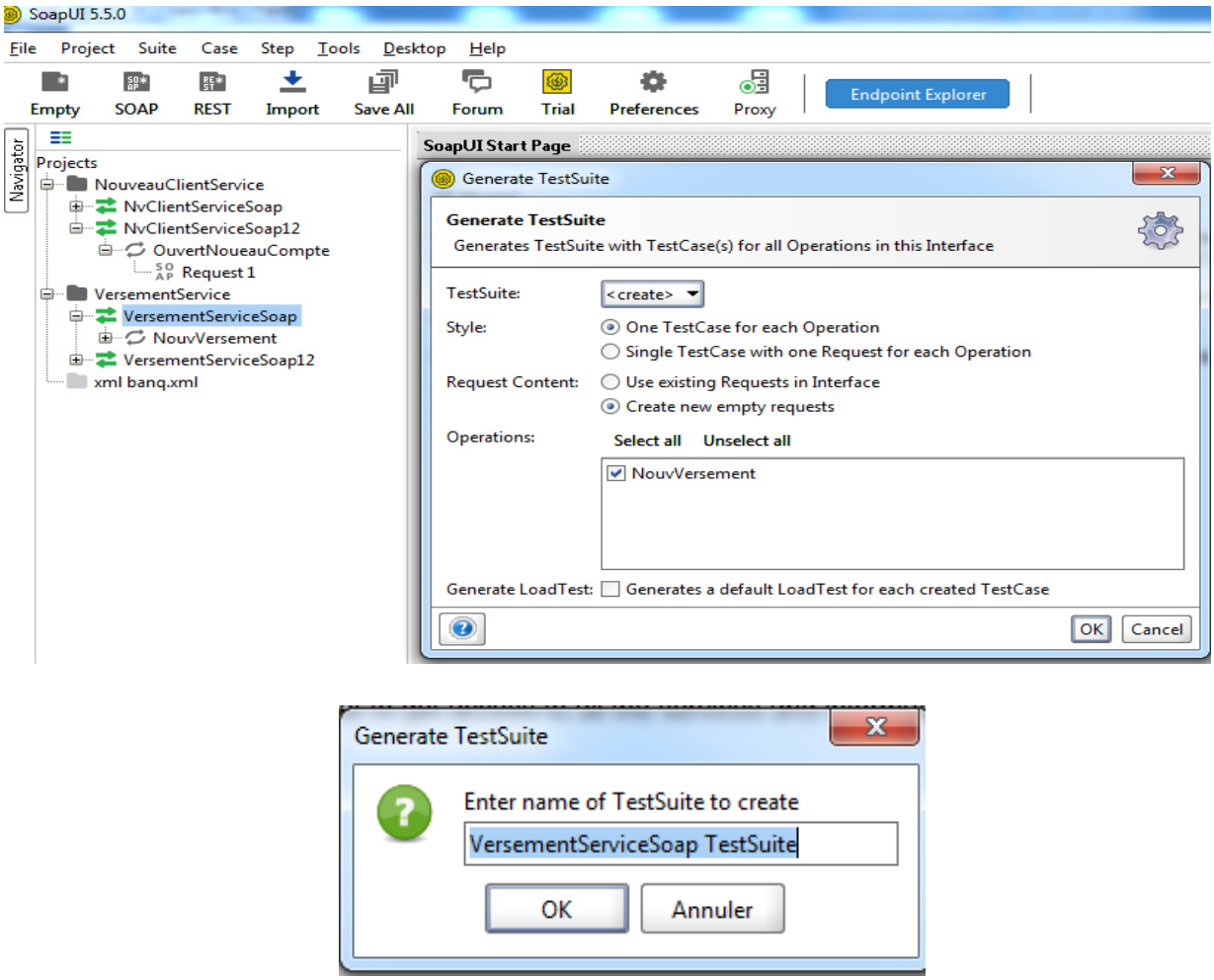


Figure 4-27 : fenêtre un nouveau projet soap



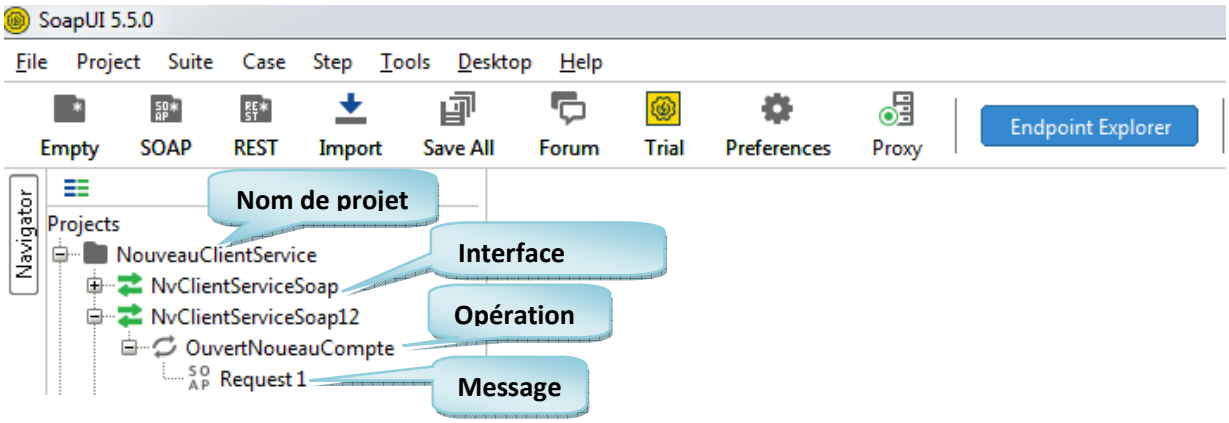


Figure 4-28 : Structure Service Web Soap

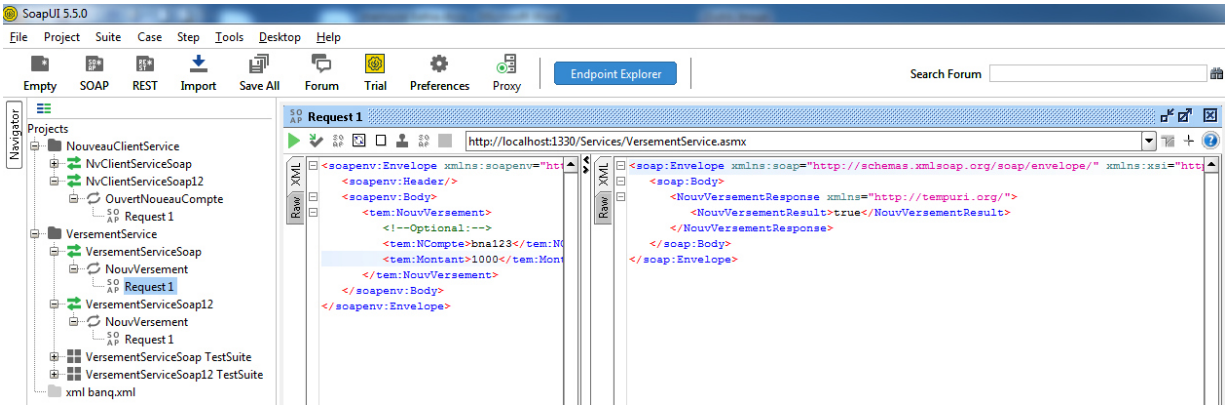
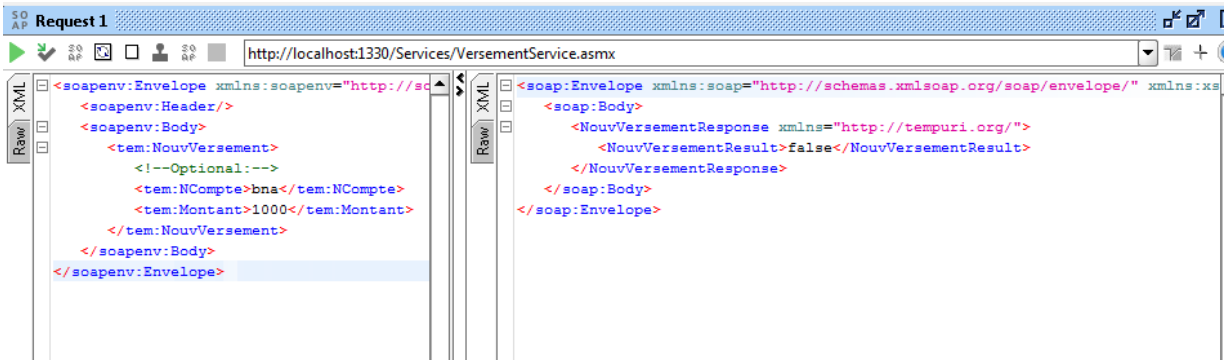
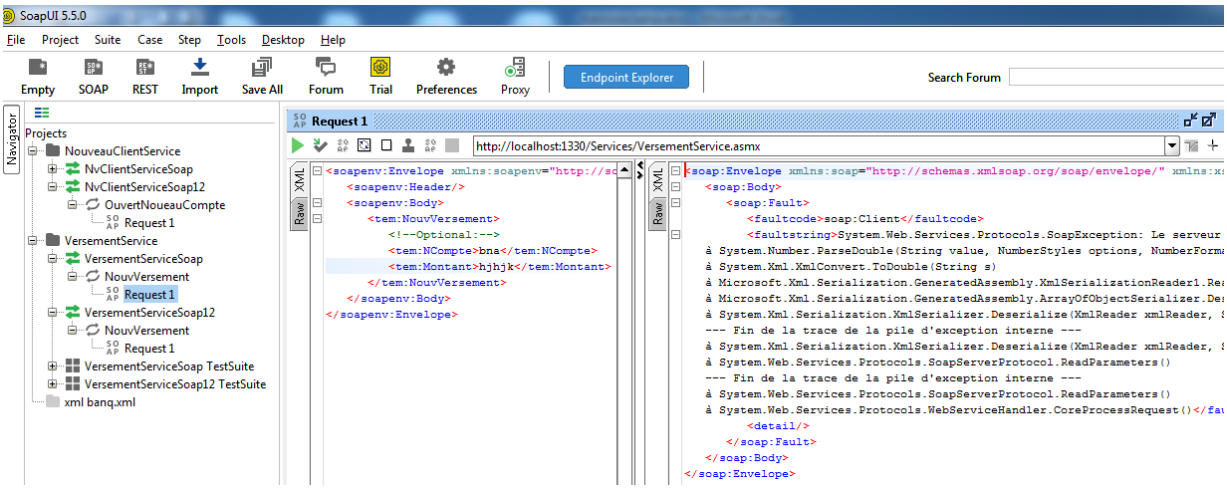


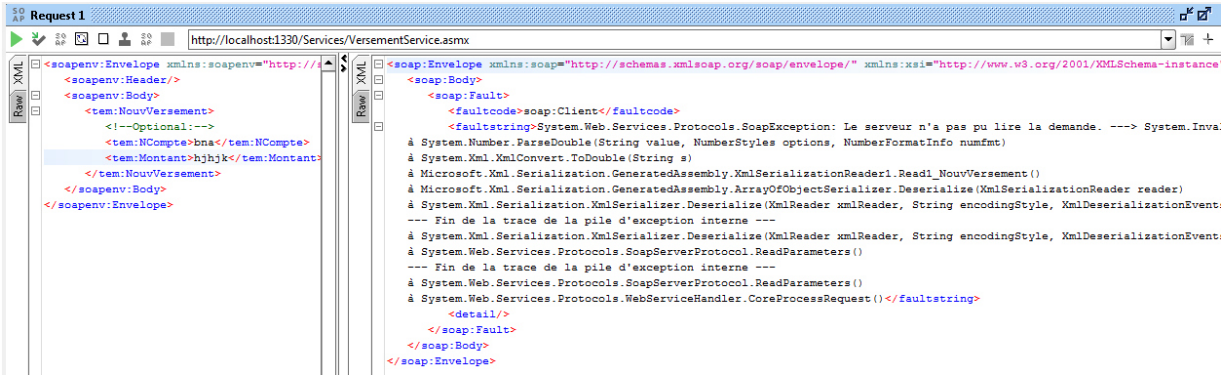
Figure 4-29 : Demande Réponse Soap

10	bna123	2021-06-26 16:...	1000,0000

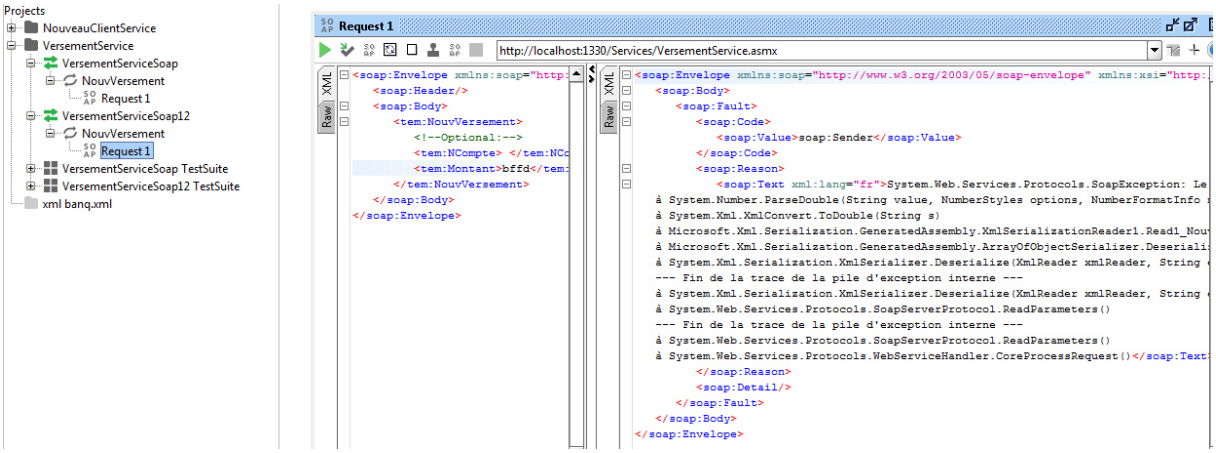
Cas erreur dans la saisir de numéro de compte







Il existe une erreur dans le document XML (7, 42). --->System.FormatException: Le format de la chaîne d'entrée est incorrect.



Résultats Empiriques :

identifiant	Brève description d'opérateur utilisé	Type de perturbation
01	Définit la valeur de *n* comme valeur limite	Perturbation de la valeur des données
02	Insérer un opérateur de paramètre dans la valeur affectée à une méthode de nœud *n*	Perturbation de la valeur des données
03	Définissez la valeur de *n* sur *null*	Valeur des données / perturbation d'interaction
04	Définissez la valeur de *n* sur "	Valeur des données / perturbation d'interaction
05	Valeur irrégulière flottante telle que 0 , 1 , −1 , + / - 1 , 5E − 324 , 1 . 7E + 308, pi, e, etc.	Perturbation de la valeur des données
06	Caractère irrégulier comme "", "," .. / "," {"," ("," ["," \ n "," \ 0 "," \ s "," \ d ")	Perturbation de la valeur des données
07	Valeur irrégulière booléenne telle que 0,1, True, False	Perturbation de la valeur des données
08	Supprimer le nœud *n* et ses nœuds enfants du message SOAP	Perturbation d'interaction
09	Échangez l'ordre des valeurs attribuées aux nœuds	Valeur des données / perturbation d'interaction
10	Échangez l'ordre des nœuds	Perturbation d'interaction
11	Modifie la longueur de la valeur affectée au nœud *n*	Valeur des données / perturbation d'interaction

Tableau 4-4: Opérateurs de mutation sur le message SOAP pour les tests des services Web [25]

A limitation de temps les tests sont effectue sur les deux services : nouveau versement et nouveau compte.

Les tests de valeur de donnes sont basé sur le tableau 8 et 9 ou les type de donnée de chaque champ et le cas de test utilisé max et min pour chacune.

Les tests de communication RPC elles sont basé sur les interactions effectuées sur l'outil soap ui et l'operateur de tableau 10.

Nombre de cas de testServices	DVP	RCP	Total
Service nouveau versement	5	6	11
Service nouveau compte	30	6	36

Tableau 4-5: Cas de test et résultats

Conclusion :

Nousavonsprésentédanscechapitrel'étudeconceptuelledenotreoutilquiprésentel'architecturegénérale,Etnousavons présentéaussilaconceptiondétailléeencommençantparlesdiagrammesetlasécuritéde lapplication,ensuitlabasededonnées.Et lestechniques utiliséespourimplémenterl'applicationconçuedanscechapitre ,ainssinousavonsessayéd'implémenterlesnotionsthéoriquedéjàmentionnéaudeuxièmechapitreainsiquenousavonsdéveloppelemodèleproposéautroisièmechapitre,enutilisantunensembled'outilsetstandard.

Onestarrivéàréaliserles différent service bancairenécessaireetlesinterfacesrequispourque notreapplicationfonctionne.

Conclusion générale :

Les manières de créer et de développer des systèmes informatiques ne cessent d'évoluer. La complexité croissante des logiciels informatiques (répartition du code, utilisation de composants externes, limitation des ressources, etc.) nécessite des méthodes de conception et de validation rigoureuses. Dans ce contexte la phase de test s'avère particulièrement importante car elle contribue à garantir un bon fonctionnement de l'implantation du logiciel, dans son environnement réel d'exécution. Cette mémoire explique la technique de Jeff Offutt&Wuzhi Xuqui expliquer laméthodesdetest par implémentation un système à base de service avec une application qui génère manuellement la technique de perturbation de donnée avec des jeux de test quiprennentencomptel'environnementSOAPdes servicesWeb pour lesvalider au mieux.

La méthode de génération de tests destinés à évaluer la robustesse d'une implantation, c'est-à-dire sa capacité à respecter certaines propriétés comportementales malgré un environnement d'exécution dégradé (susceptible de fournir des entrées incorrectes, ou d'inclure des composants externes incapables de rendre le service attendu). L'approche que nous proposons est inspirée des techniques de génération de test utilisées en test de robustesse des protocoles de communications dans lesquelles les suites de test sont générées à partir d'un modèle comportementale d'une spécification du logiciel. L'originalité de ce travail consiste à étendre cette technique pour prendre en compte un modèle de fautes (exprimant le comportement dégradé de l'environnement sous forme de mutations syntaxiques de la spécification) et un observateur. Les séquences de test produites sont alors correctes dans le sens où elles ne rejettent que des implantations non robustes vis-à-vis de cet observateur.

Travaux futurs :

Notre futur travail le plus important sur ce projet est d'automatiser la génération et l'exécution des tests.

Présentées des nouvelles techniques dans cettemémoire.

Pour généré plus de cas de test des données que les originaux pour les tests DVP et DCP

Les stratégies basées sur la perturbation des données et le combinateurtest. Sur cette base, plusieurs mutants peuvent être injectésà un moment donné pour aider à découvrir les défauts interactifs .

Bibliographie

[1]Rabhi, I., Université, O. H., & Pascal, B. (2012). *Testabilité des services Web To cite this version : HAL Id : tel-00738936 Université Blaise Pascal – Clermont II Thèse pour obtenir le grade de Docteur d ' Université Spécialité Informatique Testabilité des services Web*.

[2] https://web.maths.unsw.edu.au/lafaye/CCM/genie-logiciel/cycle-de-vie.htm.

[3] Kerdoudi, M. L. (2021). *Architectures Orientées Services*. 1–83(le¢on 2eme master).

[4]Salatge, N. (2007). *Conception et mise en oeuvre d ' une plate-forme pour la sûreté de fonctionnement des services Web To cite this version : HAL Id : tel-00135748 Par Conception et mise en oeuvre d ' une plate-forme pour la sûreté de fonctionnement des Services Web*.

[5]https://www.exoco-lmd.com/technologies-et-services-web/le-controle-de-qos-pour-les-services-web.

[6]https://download.oracle.com/otn_hosted_doc/jdeveloper/1012/web_services/ws_wsdlstructre.

[7] https://www.w3schools.com/xml/schema_intro.asp.

[8], Marcelo Arenas, Wenfei Fan et Leonid Libkin.Qu'est-ce qui est difficile avec les contraintes de schéma XML? Dans *le Conférence internationale sur les bases de données et les experts Applications de systèmes (DEXA)* , Springer, Heidelberg,2002.

[9] Boris Chidlovskii. Utilisation d'automates d'arbres réguliers comme Schémas XML. Dans *IEEE Advances in Digital Libraries 2000 (ADL 2000)* , Washington, DC, mai 2000.

[10], MakotoMurata. Automates de couverture: un modèle formel pour Schéma XML. Page Web, 2000. http://citeseer.nj.nec.com/article/murata99hedge.html (consulté en novembre 2003).

[11] Florian Reuter et Norbert Luttenberger. Cardinalité automates à contraintes: une technologie de base pour Analyseurs compatibles avec les schémas XML. Dans *le douzième Conférence internationale du World Wide Web (WWW2003)* , Budapest, Hongrie, 20-24 mai 2003. Actes TAV-WEB / ACM SIGSOFT SEN PAGE 9 Septembre 2004 Volume 29 Numéro 5.

[12]J. Offutt and W. Xu, "Generating test cases for web services using data perturbation," *ACM SIGSOFT Softw. Eng. Notes*, vol. 29, no. 5, 2004, doi: 10.1145/1022494.1022529.

[13] Bozkurt, M., Harman, M., & Hassoun, Y. (2012). *Testing and verification in service-oriented architecture : a survey*. https://doi.org/10.1002/stvr.

[14]Xu, W., Offutt, J., Luo, J., & Engineering, S. (n.d.). *Testing Web Services by XML Perturbation*

, pp. 257-266, Chicago, IL,États-Unis, novembre 2005, IEEE Computer Society.

[15] LFJ de Almeida et SR Vergilio, «Explorer les tests basés sur les perturbations pour les services Web», dansICWS '06: Actes de la conférence internationale de l'IEEE 2006 sur les services Web, pp. 717–726, Chicago,IL, États-Unis, 2006, IEEE Computer Society.

[16]S. Hanna et M. Munro, «Fault-based web services testing», dans ITGN: 5th International Conference on Information Technology: New Generations (ITNG 2008), pp. 471–476, Las Vegas, NV, USA, avril 2008,Société informatique IEEE.

[17]J. Zhang et L.-J. Zhang, «Analyse des critères et validation de la fiabilité des services Web orientés systèmes », dans ICWS '05: Actes de la Conférence internationale IEEE 2005 sur les services Web, pp. 621–628,Orlando, FL, USA, juillet 2005, IEEE Computer Society.

[18]M. Vieira, N. Laranjeiro et H. Madeira, «Benchmarking the robustness of web services», dans PRDC '07: Actes du 13e Symposium international Pacific Rim sur l'informatique fiable, pp. 322–329,Melbourne, Victoria, Australie, décembre 2007, IEEE Computer Society .

[19]E. Martin, S. Basu et T. Xie, «Tests automatisés et analyse des réponses des services Web», dans ICWS '07: Actes de la Conférence internationale 2007 de l'IEEE sur les services Web, pp. 647–654, Salt Lake City,UT, États-Unis, juillet 2007, IEEE Computer Society

[20]WT Tsai, X. WEI, Y. Chen, R. Paul et B. Xiao, «Swisscheese test case generation for web services testing », IEICE - Transactions sur l'information et les systèmes, vol. E88-D, non. 12, pp. 2691-2698, 2005.

[21]Bozkurt, M., Harman, M., & Hassoun, Y. (2010). Testing web services: A survey. *Department of Computer Science, King's College London, Tech. Rep. TR-10-01*.

[22] https://fr.wikipedia.org/wiki/Remote_procedure_call;La dernière modification de cette page a été faite le 7 avril 2020 à 01:35.

[23]https://fr.slideshare.net/Cynapsys/formation-soa

[24] https://fr.myservername.com/10-best-api-testing-tools-2021-soap

[25] hen, J., Li, Q., Mao, C., Towey, D., Zhan, Y., & Wang, H. (2014). A Web services vulnerability testing approach based on combinatorial mutation and SOAP message mutation. *Service Oriented Computing and Applications*, *8*(1), 1–13. https://doi.org/10.1007/s11761-013-0139-1.

[26]R-Michel di Scala editions Berti a Alger (Les premiers pas dans .Net Framework avec c version 2.0) Novembre 2004.

[27]Informatique, E. (2020). ´ Système Des Crédits Bancaire Base Sur La Technologie Blockchain ´.