République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ MOHAMED KHIDER, BISKRA

FACULTÉ des SCIENCES EXACTES et des SCIENCES de la NATURE et de la VIE

DÉPARTEMENT D'INFORMATIQUE

Mémoire présenté en vue de l'obtention du Diplôme :

MASTER en **Informatiques**

Option : Réseaux et Technologies de l'information et de la communication (RTIC)

Par

Yazid Gherbal

Titre :

Une approche de composition des services IOT pour l'agriculture

Membres du Comité d'Examen :

Pr.	Hamida Ammar	UMKB	Président
Pr.	ASMA BEN DAHMENE	UMKB	Encadreur
Dr.	Belaiche Hamza	UMKB	Examinatrice

Année universitaire 2021-2022

Résumé

L'Internet des objets (IoT) se compose d'appareils physiques et de composants logiciels interconnectés. Les systèmes IOT ont pour objectif de rendre l'information disponible partout et à tout moment, et fournir de nombreux services afin d'améliorer la qualité de vie et le bien-être des utilisateurs. le processus de composition sélectionne des services particuliers parmi un ensemble de services et les combine afin de former un service composite répondant au besoin complexe de l'utilisateur. Dans ce travail, nous proposons une approche de composition automatique et dynamique de services. Il répond aux exigences locales imposées par l'utilisateur en définissant un seuil de qualité de service. Au niveau global, la solution se base sur l'utilisation du principe des algorithmes génétiques (sélection, mutation, croisement et remplacement) pour avoir une solution (service composite) maximisant les valeurs de QoS. Les scénarios d'application ont montrés haute performance de l'algorithme proposé.

Mots-clés : Internet des objets, Qualité de service, Composition et sélection de services, Architecture orienté services, Algorithme génétique.

Abstract

The Internet of Things (IoT) consists of interconnected physical devices and software components. IOT systems aim to make information available anywhere and anytime, and provide many services in order to improve the quality of life and well-being of users. the composition process selects particular services from a set of services and combines them to form a composite service that meets the user's complex need. In this work, we propose a selection approach for the automatic and dynamic composition of services. It deals with local requirements imposed by the user by defining a quality of service threshold. At the global level, the solution is based on the use of the principle of genetic algorithms (selection, mutation, crossing and replacement) to have a solution (composite service) maximizing the QoS values. The application scenarios showed high performance of the proposed algorithm.

Keywords : Internet of things, Quality of service, Composition and selection of services, Service-oriented architecture, Genetic algorithm.

Dédicace

Je dédie ce modeste travail :

A la mémoire de mon très chèr père qui a toujours souhaité voir ce jour,

A ma très chère maman qui m'a toujours soutenu et encouragé,

A Ma chère épouse **Hanane**, qui m'a encouragé,

A mes chers enfants **Lina**, **Adam**, **Layane,**

A mes frères adorés, leurs femmes et leurs enfants,

A toutes ma famille et **A** tous mes amis(es),

Enfin, **A** toutes les personnes qui nous ont apporté de l'aide.

Yazid gherbal

REMERCIEMENTS

Nous tenons dans un premier temps à remercier le Dieu tout puissant qui nous a donné le courage et la volonté pour mener à bien ce modeste travail.

Ce mémoire n'aurait jamais pu voir le jour sans le soutien actif d'un certain nombre de personnes que nous tenons à remercier, toutes celles et ceux qui ont contribué à la réalisation de ce modest travail :

En particulier mon ami **Youcef Refrafi** qui m'a aidé pendant la préparation de ce travail malgré ses occupations

Mon encadreur Madame **Ben dahmene Asma**

pour avoir encadré et conseillé dans la réalisation de ce travail.

Ma chère épouse **Dr.Hanane Ben Gherbal**, qui m'a encouragé et soutenu tout au long de cette période

les membres du jury qui ont accepté d'évaluer notre travail.

Dr. Souraya HAMIDA du Laboratoire LINFI, Université Batna pour ses précieux conseils

L'expert agricole **Refrafi toufik**, et l'ingénieur agronome, **Binazrine Mohammed** pour les informations et l'assistance.

Table des matières

Remerciements ii

Table des matières iii

Table des figures vii

Liste des tables ix

Liste des algorithmes x

Liste des abréviations xi

Introduction générale 1

1 Etat de l'art 4

1.1 Introduction 4

1.2 Internet des objets 5

1.2.1 Définition de l'internet des objets 5

1.2.2 Capteurs et Actionneurs 6

1.2.3 Axes de l'internet des objets 7

1.2.4 Caractéristiques d'un système d'Internet des objets 7

1.2.5 Architecture d'un système IOT 8

1.2.5.1 Couche de Perception 9

1.2.5.2 Couche de Transmission 9
1.2.5.3 Couche Application 10
1.2.6 Internet des objets comme un système 10
1.2.6.1 Internet des objets comme un système de système 10
1.2.6.2 Réseaux et Internet des objets 10
1.2.6.3 Internet des objets et les données 11
1.2.6.4 Internet des objets et les nouveaux services 11
1.2.7 Domaines d'application 11
1.3 L'architecture orienté service (SOA) 13
1.3.1 La notion de service 13
1.3.2 La notion de SOA 14
1.3.3 Caractéristiques de l'architecture SOA 14
1.3.4 Acteurs de l'architecture SOA 14
1.4 Composition des services IOT 15
1.4.1 définitions de la composition de services 15
1.4.2 Etapes de la composition des services 16
1.4.2.1 Spécification abstrait 16
1.4.2.2 Génération de service composite 17
1.4.2.3 Evaluation des services composites 17
1.4.2.4 Déploiement et exécution des services 17
1.4.3 Composition basé algorithmes génétiques 18
1.4.4 Approches existants pour la composition des services IOT 21
1.5 Conclusion 24

2 Une approche de composition de services IOT basée QoS 25

2.1 Introduction 25

2.2 Motivation 26
2.3 Architecture générale du système 26
2.3.1 Architecture proposée : 26
2.4 Architecture détaillée 28
2.4.1 Modélisation des services 34
2.4.1.1 Service abstrait 34
2.4.1.2 Service concret 35
2.4.2 Qualité de service 36
2.4.2.1 Energie 36
2.4.2.2 Localisation 36
2.4.2.3 Temps de réponse 36
2.4.2.4 Fiabilité 37
2.4.2.5 Disponibilité 38
2.4.2.6 Prix 39
2.4.2.7 Réputation 39
2.4.3 Mise à l'échelle de qualité de service 40
2.5 Processus de composition de service IOT 42
2.5.1 Phase de sélection locale 42
2.5.2 Phase de sélection globale 44
2.5.2.1 Fonction d'agrégation de QoS 44
2.5.2.2 Les paramètres de l'AG 44
2.5.2.3 Description des opérateurs génétiques 46
2.5.3 Phase de déploiement et exécution des services 47
2.5.4 Phase de contrôle du plan d'exécution 47
2.6 Scénario d'application 48

2.6.1 IOT dans le secteur agricole 48
2.6.2 Capteurs utilisés dans le scénario 49
2.6.3 Actionneurs utilisés dans le scénario 50
2.6.4 Description du scénario 50
2.7 Conclusion 52

3 Implémentation et résultats **53**
3.1 Introduction 53
3.2 langage et les outils de développements : 53
3.2.1 Le langage JAVA 53
3.2.2 Netbeans 54
3.2.3 MySQL 54
3.3 Description du système 55
3.3.1 Structuration de l'algorithme de composition 55
3.3.1.1 Extrait de code les classe principale 56
3.3.2 Présentation de l'application 59
3.4 Expérimentation 63
3.5 Conclusion 65

Conclusion générale et perspectives **66**

Bibliographie **67**

Table des figures

1.1 Les dimensions de l'IOT[9]. 7

1.2 Architecture à trois couches de l'IoT 9

1.3 Domaines d'application de l'IoT[13] 12

1.4 Composition de services 16

2.1 Architecture proposée[26] 27

2.2 Modèle de traitement de composition de services IOT 29

2.3 Formule de la requête 29

2.4 Processus de composition de service par diagramme de séquence 33

2.5 Représentation d'un service abstrait 34

2.6 Représentation d'un service concret. 35

2.7 Organigramme de sélection globale de services 47

2.8 Scénario applicatif irrigation : les services abstraits 50

2.9 Scénario applicatif Fertilisation : les services abstraits 51

3.1 Extrait de code : classe Macthing 56

3.2 Extrait de code : classe plan de composition 57

3.3 Extrait de code : classe Algorithme génétique 58

3.4 Méthodes de L'algorithme génétique 59

3.5 fenêtre principale de projet 60

3.6 Formulation de requet utilisateur 61

3.7 Plan de Composition des services abstrait " Irrigation " 62

3.8 Plan de Composition des services abstrait " Fertialisation " 63

3.9 Changement Fitness dans tous les génerations de AG 64

3.10 Changement L'energie avant et aprés le processus de composition 65

Liste des tableaux

1.1 Tableau de comparaison des approches étudiées 23

2.1 les domaines de valeurs de critères de QoS 42

2.2 Fonction d'agrégation de QoS 44

2.3 description des capteurs IOT existants dans la serre agricole 49

2.4 description des Actionneur IOT existants dans le serres agricole 50

2.5 description des sorties des services abstraits "irrigation" 51

2.6 description des sorties des services abstraits "Fertilisation" 52

Liste des Algorithmes

1 Algorithme de matching 30

2 Algorithme de Calcul du temps de reponse d'un service concret 37

3 Algorithme de Calcul de la fiabilite d'un service concret 38

4 Algorithme de Calcul de la disponibilite d'un service concret 39

5 Algorithme de Calcul de la reputation d'un service Abstrait 40

6 Algorithme de Sélection locale des services concrets 43

7 Algorithme d' évaluation de la population t 46

Acronymes

Les différentes abréviations et notations utilisées tout au long de ce mémoire sont expliquées ci-dessous :

QoS	**Q**uality **o**f **S**ervice
SOA	**S**ervice **O**riented **A**rchitecture
BPEL	**B**usiness **P**rocess **E**xecution **L**anguage
SC	**S**ervice **C**oncret
SA	**S**ervice **A**bstrait
PDA	**P**ersonal **D**igital **A**ssistant
RFID	**R**adio **F**réquence **Id**entification
IOT	**I**nternet **o**f **T**hings
MDP	**M**écanisme de **D**éveloppement **P**ropre
AG	**A**lgorithme **G**énétique
SQL	**S**tructured **Q**uery **L**anguage

Introduction générale

L'Internet des objets représente une infrastructure qui remplit le gap entre les capteurs simples qui fournit des données bruts, imparfaites et la virtualisation basée sur des applications de haut niveau qui fournissent des services sophistiqués.

De nos jours, l'IoT devient de plus en plus répandu et adapté aux interactions des utilisateurs, qui composent les technologies de service de manière hétérogène. En règle générale, les appareils IoT sont intégrés de manière hétérogène et dynamique avec les fonctions de fiabilité liées à leur QoS pour chaque service atomique. Chaque service atomique est lié au dispositif IoT qui fournit la fonctionnalité du service. Dans la plupart des cas, un seul service n'est pas suffisant pour répondre aux exigences complexes de l'utilisateur.

De plus, un seul service IoT ne peut pas satisfaire les demandes de l'utilisateur composé, car certains des services complexes sont offerts par un ensemble d'appareils IoT. Dans ce cas, un service composite est nécessaire. Un service composite orchestre un ensemble de services uniques pour résoudre avec succès un objectif complexe d'une manière qui ajoute de la valeur aux services composites fournis.

La composition de services permet donc à l'utilisateur d'employer les services disponibles dans son environnement pour résoudre des requêtes complexes, tout en considérant les propriétés non fonctionnelles des services et les informations contextuelles pour répondre mieux à l'utilisateur. De ce fait, la composition de services est un processus complexe faisant intervenir un ensemble d'étapes intermédiaires telle que la sélection pour pouvoir choisir le meilleur service qui participera à la composition, ou sélectionner le meilleur service composite.

Problématique

Pour répondre au besoin complexe de l'utilisateur qui cherche à obtenir le bon service au bon moment et au bon endroit ;

Pour faire face aux problèmes posés par la composition on doit fixer des contraintes de sélection (locale et globale) afin de permettre au système de composition d'élire le service le plus approprié. Plus précisément, la sélection de services compare les services fonctionnellement équivalents en se basant sur leurs performances non-fonctionnelles comme par exemple, le temps d'exécution, la disponibilité, la fiabilité, le niveau d'énergie, le temps de réponse, etc.

L'objectif de ce travail est de développer un mécanisme de composition de service qui permet de choisir le meilleur service parmi les services disponibles selon la qualité de service et satisfaire la requête de l'utilisateur en utilisant les algorithmes génétiques.

Dans ce travaille nous présentons la définition du concept et l'architecture de l'IOT.

Organissation du mémoire

Après l'introduction générale, ce mémoire réparti trois chapitres fini par une conclusion générale

Chapitre 1 Etat de l'art :

Ce chapitre est consacré aux concepts de base : Internet des Objets,L'architecture orienté service (SOA) ,composition des services IOT, Nous terminons ce chapitre par l'exposition de certaines approches existantes dans la littérature.

chapitre 2 Composition de services IOT basés sur les algorithmes génétiques :

nous présentons les algorithmes génétiques, puis nous détaillons notre contribution qui consiste en une approche de sélection pour la composition de services.

Un scénario d'application (serres agricole intelligent) illustrant la composition de services est également présenté dans ce chapitre.

Chapitre 3 Implementation et résultats :

Le dernier chapitre, concerne la mise en œuvre de notre approche, en commenceront par la description des outils et l'environnement de développement. Ensuite, nous présentons

quelques interfaces qui montrent les résultats obtenus d'après l'implémentation de notre modèle.

Nous cloturerons ce mémoire par une conclusion et quelques perspectives.

Chapitre 1

Etat de l'art

1.1 Introduction

L'Internet des objets (IOT) est une révolution technologique en informatique et en communication qui a mobilisé le domaine de l'agriculture ces dernières années. Dans ce premier chapitre, nous allons survoler la notion d'Internet des objets.

Comme premier titre, nous allons présenter les concepts relatifs cette nouvelle technologie qui a pu marquer son utilisation presque dans tous les domaines tel que l'agriculture ;

nous définissons l'IOT comme un système en relation avec le réseau, les données et les nouveaux services. Puis nous allons voir l'architecture des IOT avant de passer, finalement, aux différents domaines d'applications et les problèmes de l'IOT.

Comme deuxième titre nous allons détailler l'architecture orienté service (SOA) ainsi que les caractéristique précisément leur processus de composition et intégration. Finalement, nous allons classer quelques travaux connexes.

1.2 Internet des objets

1.2.1 Définition de l'internet des objets

Il y a plusieurs définitions du concept Internet des Objets (en anglais : Internet of Things ou IoT). Dans ce qui suit, on présente quelques définitions.

Définition 1 :

L'objet connecté est un objet électronique qui peut transmettre des informations en temps réel via une liaison sans fil à un autre dispositif connecté. Ces informations peuvent être de plusieurs types [1].

Un objet peut être une entité physique ou virtuelle ayant des identités et des personnalités virtuelles, opérant dans des espaces intelligents et utilisant des interfaces intelligentes pour se connecter et communiquer au sein de contextes d'usages variés [2].

Définition 2 :

L'Internet des objets peut être défini aussi comme étant un réseau de réseaux qui permet, via des systèmes d'identification électroniques normalisés et unifiés, et des dispositifs mobiles sans fil, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi, de pouvoir récupérer, stocker, transférer et traiter les données sans discontinuité entre les mondes physiques et virtuels [3].

Définition 3 :

L'Internet des objets est une extension de l'Internet actuel envers tout objet pouvant communiquer de manière directe ou indirecte avec des équipements électroniques eux-mêmes connectés à l'Internet [4].

Dans le cadre de notre travail, nous adoptons la définition proposée dans [5] qui est :

"L'Internet des Objets (IOT) se définit comme un réseau mondial de services interconnectés et d'objets intelligents de toutes natures destinés à soutenir les humains dans les activités de la vie quotidienne grâce à leurs capacités de détection, de calcul et de communication. Leurs aptitudes à observer le monde physique et à fournir des informations pour la prise de

décision, seront partie intégrante de l'architecture de l'Internet du futur".

Les capteurs et actionneurs forment les éléments clés de l'internet des objets. Ils suivent l'état de leur environnement, obtiennent des informations sur la température, le mouvement, la position, etc. Ils constituent un réseau généralement composé d'un nombre potentiellement élevé de nœuds. Ces capteurs doivent faire face à de nombreux problèmes de communication comme la sécurité et confidentialité, leur mobilité, leur courte portée, leur fiabilité, leur robustesse, leur évolutivité et leurs ressources (énergétiques, capacité de stockage et de traitement limitées, bande passante, etc.).[6]

1.2.2 Capteurs et Actionneurs

Les capteurs sont les yeux et les oreilles de l'objet IoT pour voir son environnement, et les actionneurs sont les jambes et les mains qui remplissent ses fonctions [7] :

1. **Capteur** : Un capteur est un appareil qui convertit un paramètre physique en une sortie électrique. Un capteur est un type de transducteur. Il disposant de capacités de mesures, voire d'actions, sur leur environnement La température, le taux d'humidité, la luminosité ambiante, la détection de présences ou de mouvements via un accéléromètre, présence de gaz, de polluants ou encore la géolocalisation font partie des informations les plus couramment collectées sur ce type de matériel.
2. **Actionneur** : Un actionneur est un dispositif qui convertit un signal électrique en sortie physique, c'est-à-dire un mouvement. Un actionneur peut être contrôlé par la tension ou le courant électrique, la pression pneumatique ou hydraulique, ou même la puissance humaine. Dans les systèmes embarqués, les actionneurs sont principalement contrôlés par l'électricité. Lorsque le signal de commande est reçu, l'actionneur convertit l'énergie électrique en mouvement mécanique.

1.2.3 Axes de l'internet des objets

L'IOT ajoute une dimension "objet" aux deux déjà existantes (temporelle et spatiale), désormais, un objet peut communiquer avec n'importe quel objet (ou personne), à n'importe quelle heure et à n'importe quelle place [8].

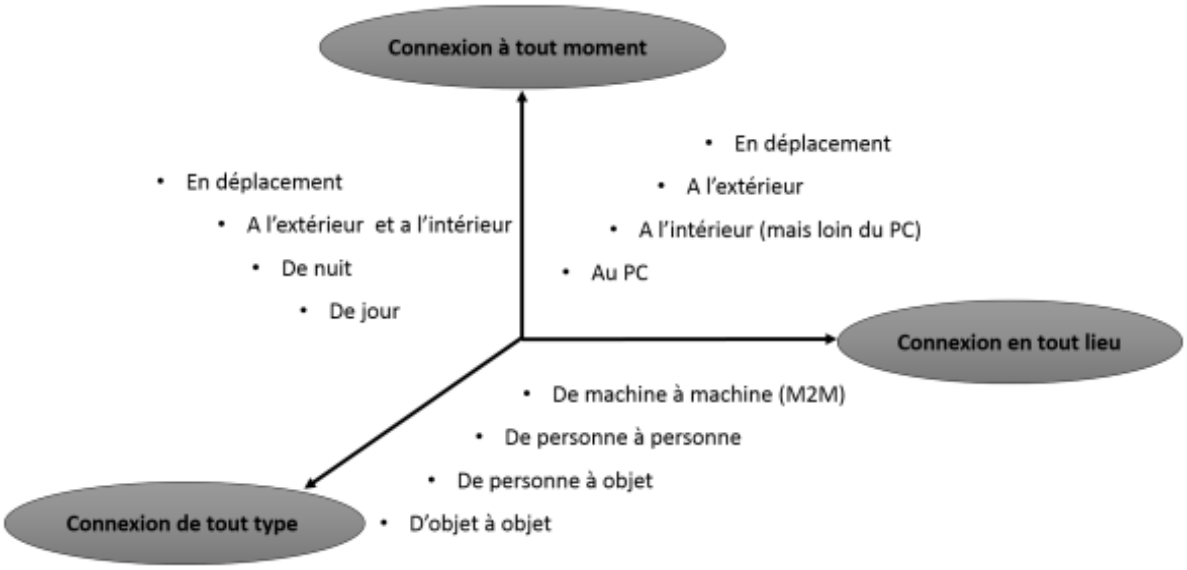

FIG. 1.1 – Les dimensions de l'IOT[9].

1.2.4 Caractéristiques d'un système d'Internet des objets

Les caractéristiques fondamentales de l'IOT sont les suivantes [10] :

- **Inter-connectivité** : tout peut être interconnecté avec l'infrastructure globale d'information et de communication
- **Services liés aux objets connectés** : L'IOT est capable de fournir des services liés aux objets dans les limites des contraintes, telles que la protection de la vie privée et la cohérence sémantique entre les choses physiques et leurs objets virtuels associés. Afin de fournir des services liés aux choses (i.e. objets) dans les contraintes sur les choses, les technologies dans le monde physique et le monde de l'information vont changer.
- **Hétérogénéité** : Les périphériques de l'IoT sont hétérogènes en fonction des plateformes

matérielles et des réseaux. Ils peuvent interagir avec d'autres appareils ou plates-formes de services via différents réseaux.

- **Changements dynamiques :** L'état des dispositifs change dynamiquement, par exemple dormir et se réveiller, être connecté et / ou déconnecté ainsi que le contexte des dispositifs, y compris l'emplacement et la vitesse. de plus, le nombre d'appareils peut changer de façon dynamique.
- **Énorme échelle** : Le nombre de périphériques qui doivent être gérés et qui communiquent entre eux sera d'au moins un ordre de grandeur supérieur à celui des périphériques connectés à Internet. Encore plus critique sera la gestion des données générées et leur interprétation à des fins d'application. Cela concerne la sémantique des données, ainsi que la gestion efficace des données [11].
- **Sécurité :** Comme nous gagnons des avantages de l'IoT, nous ne devons pas oublier la sécurité. En tant que créateurs et destinataires de l'IoT, nous devons concevoir des mécanismes assurant la sécurité. Cela inclut la sécurité de nos données personnelles et la sécurité de notre bien-être physique. La sécurisation des points de terminaison, des réseaux et des données qui les traversent signifie la création d'un paradigme de sécurité qui évoluera.
- **Connectivité :** La connectivité permet l'accessibilité et la compatibilité du réseau ; l'accessibilité se met sur un réseau alors que la compatibilité fournit la capacité commune de consommer et de produire des données [12].

1.2.5 Architecture d'un système IOT

Dans l'IoT, chaque couche est définie par ses fonctions et les périphériques utilisés dans cette couche. L'architecture de l'IOT est généralement divisée en trois couches, la couche perception, la couche réseau et la couche application [13].

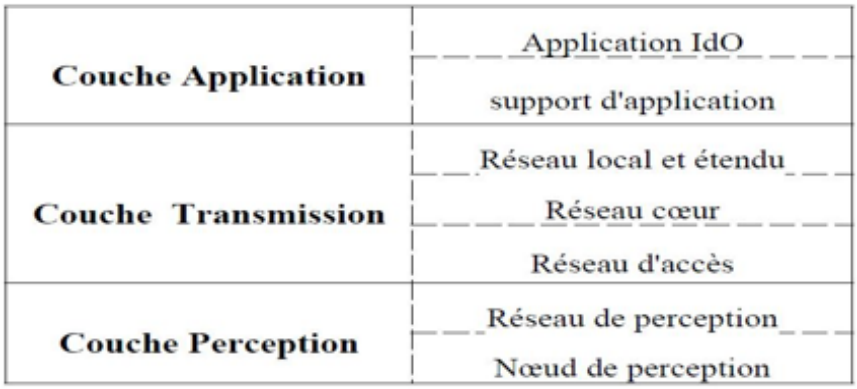

FIG. 1.2 – Architecture à trois couches de l'IoT

1.2.5.1 Couche de Perception

La couche de perception (peut être appelé "couche de périphérique", "couche sensorielle" ou "couche de reconnaissance") qui est la couche la plus basse de l'architecture IoT, est responsable de la capture des informations du monde réel et leur représentation au format numérique. Elle inclut les technologies utilisées pour la détection (collecte des données de l'environnement), l'identification (identification d'objets), l'activation (réalisation données détectées) et la communication (établissement de la connectivité entre appareils intelligents hétérogènes) avec un minimum d'interaction humaine. Selon les fonctionnalités qu'elle assure,

Cette couche peut être divisée en deux sous-couches : les noeuds de perception (ou nœuds sensoriels) et le réseau de perception (comme réseau des capteurs).

1.2.5.2 Couche de Transmission

La couche de transmission (appelée aussi «couche de transport» ou «couche réseau») est responsable de transmission des données collectées par les noeuds de perception à l'unité de traitement de l'information (ou unités de prise de décision de haut niveau) à travers un réseau ou une interconnexion des réseaux. Cette couche permet alors une intégration d'une variété de réseaux, de technologies et de protocoles hétérogènes.

Cette couche peut être divisée en trois sous-couches : réseau d'accès, réseau cœur et réseau local et étendu.

1.2.5.3 Couche Application

C'est la couche la plus haute de l'architecture IOT visible par l'utilisateur final. La couche application a pour but de gérer et de fournir les applications globales en se basant sur la les informations collectées par la couche de perception. Elle fournit aux utilisateurs finaux un accès aux services personnalisés sur le réseau, en fonction de leurs besoins, grâce à l'utilisation de divers appareils mobiles et équipements terminaux.

Cette couche peut être divisée en deux sous-couches : couche de support d'application et applications IOT.

1.2.6 Internet des objets comme un système

L'Internet des objets est en relation avec le réseau, les données et les nouveaux services :

1.2.6.1 Internet des objets comme un système de système

L'IOT désigne plutôt diverses solutions techniques (RFID, TCP/IP, technologies mobiles, etc.) qui permettent d'identifier des objets, de capter, stocker, traiter, et transférer des données dans les environnements physiques mais aussi entre des contextes physiques et des univers virtuels.Le principal défi n'est pas tant d'inventer de nouvelles technologies que de maîtriser, connecter et intégrer celles qui existent déjà.[15]

1.2.6.2 Réseaux et Internet des objets

L'IOT se compose d'un ensemble hétérogène de réseaux qui permettent la communication de ces objets. Parmi les plus connus, les réseaux cellulaires des opérateurs télécoms qui permettent aux objets équipés d'une carte SIM M2M de remonter et envoyer les données. En plein émergence, les réseaux LPWA, avec notamment LoRa et Sigfox. Réseaux bas débit longue portée, ce sont des protocoles entièrement dédiés aux communications entre objets [14].

1.2.6.3 Internet des objets et les données

Le petit élément central de l'Internet des objets est constitué de données. Surtout, la capacité de capter des données brutes, température, vibration, humidité... pour les transformer en informations intelligentes et exploitables. Ces milliards d'objets connectés vont créer une énorme quantité de données qu'il faudra stocker, analyser, sécuriser et restituer pour divers usages. Un enjeu de taille donc.

1.2.6.4 Internet des objets et les nouveaux services

La valeur ajoutée apportée par l'Internet des Objets est dans les nouveaux usages que cela va amener. Dans le secteur de l'industrie par exemple on peut désormais surveiller les machines à distance, faire de la maintenance prédictive des équipements, ou améliorer la traçabilité des produits.

Chaque jour, les objets connectés vont générer des milliards d'informations qui permettront aux entreprises de créer de nouveaux services.

1.2.7 Domaines d'application

Les applications potentielles de l'IOT sont nombreuses et variées, pénétrant dans pratiquement tous les domaines de la vie quotidienne des individus, des entreprises et de la société dans son ensemble.

En fonction de leurs fonctionnalités, les applications IOT peuvent être divisées en trois catégories :

- **Applications de collection d'informations** : elles sont chargées de collecter les données des nœuds de perception et de leur stockage local.
- **Applications d'analyse** : elles sont concernées par le prétraitement hors ligne des données collectées pour créer un modèle générique à utiliser pour l'évaluation de futures données à collecter ultérieurement.

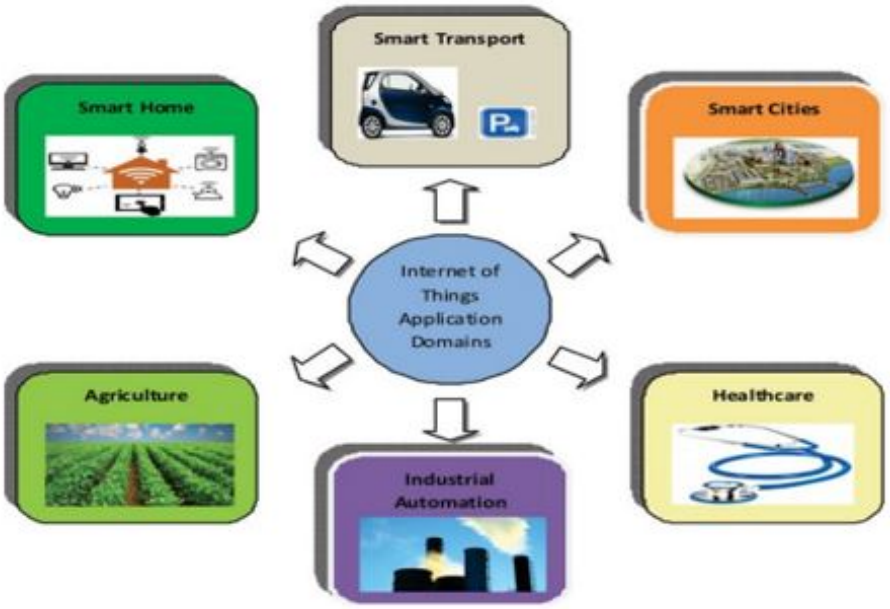

FIG. 1.3 – Domaines d'application de l'IoT[13]

– **Applications prise de décision en temps réel** : elles sont impliquées dans la prise des mesures et actions appropriées en fonction des données capturées et analysées.

L'IOT possède un vaste domaine d'applications, comme illustré à la figure 1.3 en relation avec des demandes de l'industrie. Celles-ci comprennent des applications destinées aux consommateurs, telles que les appareils portables, les maisons intelligentes (Smart Home) et les soins de santé intelligents (Healthcare) ; applications commerciales telles que la logistique et la vente au détail ; applications industrielles telles que la gestion des ressources et de l'énergie, le transport intelligent (Smart Transport) et la fabrication (industrial automation) ; et des applications spécifiques au secteur public telles que les villes intelligentes (Smart cities), la sécurité et la surveillance, etc., qui visent à améliorer la qualité de la vie humaine.

Les applications et les services IoT sont directement accessibles aux utilisateurs via l'utilisation de divers appareils de poche tels que les téléphones portables, les ordinateurs, les assistants numériques personnels (PDA), etc.

1.3 L'architecture orienté service (SOA)

1.3.1 La notion de service

Un service est un mécanisme capable de fournir une ou plusieurs fonctionnalités. Un service peut être utilisé conformément aux restrictions et règles définies par le fournisseur et via une interface. Les services pour environnements intelligents sont généralement considérés comme des extensions de services Web existants (par exemple, messagerie, services de localisation, gestion d'appareils, applications multimédias...) tout en reprenant les principes de base. On distingue deux catégories de prestations : Les services utilisateurs et les services logiciels [16].

Il existe plusieurs définitions, dont [17] :

Un service est une brique logicielle autonome qui fournit une fonction bien définie tel que l'analyse d'informations, la recherche d'informations, etc.

Un service est considéré comme une entité logicielle autonome dotée d'une interface bien définie qui peut être accessible sans aucune connaissance de sa technologie sous-jacente. Un service est autonome dans la mesure où il peut se suffire à lui-même, toutefois il peut être publié et rendu disponible pour être utilisable par des tiers. Ainsi des services peuvent être utilisés tels quels ou bien être composés pour mener à terme un processus complexe et atteindre un objectif précis de plus haut niveau.

Un service peut être aussi défini comme étant un module qui peut être invoqué, qui est assigné à une fonction spécifique, et qui offre une interface bien définie [17]. Un service est décrit par son descripteur de service, sa spécification en quelque sorte. Le descripteur comporte des :

- **Informations fonctionnelles :** la sémantique des opérations, le comportement du service (pré condition, post-condition, invariant, exception, propriétés), l'interface du service.
- **Informations non fonctionnelles :** prix, politique, spécification des mesures de la qualité de service, information de déploiement, etc.
- **Informations additionnelles :** composées d'informations sur le service, non spécifiées par le fournisseur du service telles que les notes, les rapports d'utilisation, etc.

1.3.2 La notion de SOA

Une architecture de services permet de standardiser l'accés aux ressources et/ou aux fonctionnalités des objets communicants en les représentants sous formes de services.**SOA** est une architecture de conception qui est définie par un ensemble de principes de conception [19]. Un service est défini par un contrat (appelé aussi interface) qui est une spécification abstraite de ses fonctionnalités. Ce contrat décrit : (i) ce que le service fournit, (ii) comment y accéder et (iii) éventuellement, quelles sont ses propriétés non fonctionnelles.

1.3.3 Caractéristiques de l'architecture SOA

L'architecture orienté service est caractérisée par :

• Un couplage faible entre les services : implique qu'un service n'appelle pas directement un autre service.

• La réutilisation de service.

• L'indépendance par rapport aux aspects technologiques : c-à-d les services sont indépendants des plates-formes.

• La découverte des services disponibles.

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

1.3.4 Acteurs de l'architecture SOA

L'architecture orienté service est basée sur trois acteurs principaux définis comme suit :

• **Service fournisseur** : On peut l'appeler fournisseur il met le service en application et il le rend disponible pour tout le monde sur internet.

• **Service consommateur** : C'est un client demandeur ou un consommateur qui fait la demande d'un service bien précis pour répondre à ses besoins.

• **Service annuaire** : Le registre fournit un endroit où le consommateur peut trouver des

nouveaux services et le fournisseur dispose une description pour des nouveaux services.

1.4 Composition des services IOT

Dans cette partie nous présentons des définitions préliminaires, ensuite nous détaillons les différents travaux qui proposent des approches de composition de service présents dans la littérature. est décrit suivi d'une étude comparative des différentes solutions.

1.4.1 définitions de la composition de services

Définition 1 :

La composition de services a été définie comme étant la capacité d'offrir des services à valeur ajoutée en combinant des services existants offerts par différentes organisations.

Définition 2 :

Elle a été définie également comme étant une technique permettant d'assembler des services pour réaliser un objectif particulier, par l'intermédiaire de primitives de contrôles (boucle ,test , traitement d'exception ,etc) et d'échange (envoi et réception de message)[21].

Définition 3 :

La composition de services spécifie quels services ont besoin d'être invoqués, dans quel ordre, quelles sont les données à échanger, et comment traiter les situations d'exceptions. La composition de services peut être vue comme un mécanisme qui permet l'intégration des services dans une application.

La troisième définition est la plus référencée dans la littérature. Nous constatons que les différentes définitions s'accordent sur le fait que la composition de services vise la création de nouveaux services, offrant de nouvelles fonctionnalités, à partir des services existants. Le nouveau service résultat d'une composition de services est appelé service composite. Son exécution nécessite alors l'invocation de plusieurs autres services afin de faire appel à leurs

fonctionnalités. Les services invoqués lors d'une composition de services sont appelés services composites [22].

La composition de services peut être vue comme un mécanisme permettant l'intégration des services pour réaliser une application. Le résultat d'une composition peut être un nouveau service, appelé service composite.

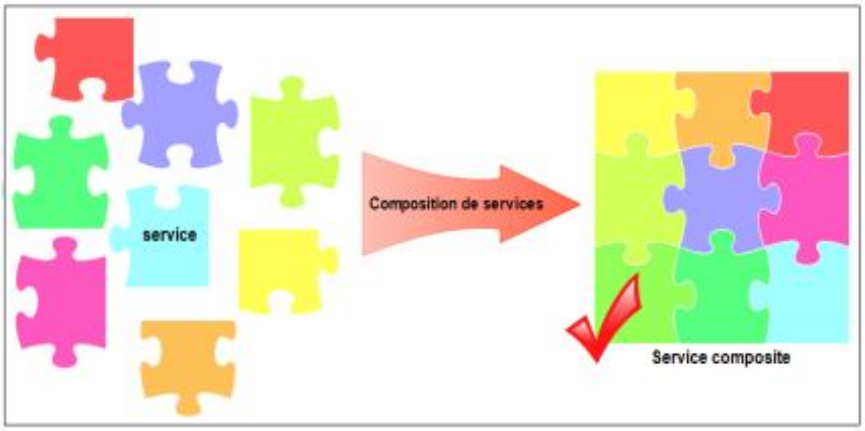

FIG. 1.4 – Composition de services

1.4.2 Etapes de la composition des services

La réalisation d'une application par composition de services comporte plusieurs étapes qui permettent le passage incrémental d'une spécification abstraite vers une composition concrète de services, c'est à dire une composition prête à être exécutée[22].

1.4.2.1 Spécification abstrait

Lors de cette phase, les fournisseurs de services décrivent et annonce leur services dans des annuaires (registre). Les attributs fonctionnels (entrées/sorties) sont nécessaires lors de la combinaison et l'invocation des services tandis que les attributs non-fonctionnels (qualité de services) sont utilisés pour évaluer les services. Cette phase consiste également à identifier les fonctionnalités qui doivent être remplies par l'application résultante de la composition de

services. Ces fonctionnalités peuvent êtres spécifiées sous forme d'une requête qui exprime les besoins de l'utilisateur.

1.4.2.2 Génération de service composite

Cette phase consiste à identifier les services nécessaires à la composition afin de répondre aux besoins fonctionnels identifiés au préalable.

Le générateur de composition (ou planificateur) tente de répondre à ces besoins en combinant, selon une stratégie donnée, les services disponibles. Le générateur prend souvent les fonctionnalités des services disponibles comme entrées. Les services composites ainsi générés représentent des candidats pour la réalisation de la composition concrète de services.

1.4.2.3 Evaluation des services composites

Il arrive souvent que plusieurs services possèdent des fonctionnalités similaires. Il est donc possible que le planificateur génère plusieurs plans de composition (services composites) qui satisfassent les besoins fonctionnels de l' application. Cette phase sélectionne alors, selon une stratégie donnée, un service composite parmi les candidats précédemment identifiés. La méthode la plus communément utilisée est les fonctions d'utilités. Dans ce cas, les services composites sont évalués en se basant sur les informations fournies par les attributs non fonctionnels des services composants. Les services sélectionnés sont préparés pour l'exécution : les services sont configurés et d'éventuelles adaptations sont réalisées.

1.4.2.4 Déploiement et exécution des services

Lors de cette phase, les services sélectionnés et préparés au préalable sont déployés sur les plates-formes d'exécution. Il est ainsi possible d'invoquer ces services pour réaliser correctement et concrètement leur composition.

L'exécution d'un service composite peut être vue comme une séquence de messages échangés selon le modèle de composition. Le flot de données du service composite est défini par le

transfert des données de sortie d'un service composant aux prochains services composants qu'il précède directement dans le service composite.

1.4.3 Composition basé algorithmes génétiques

Les algorithmes génétiques (notés *AG*) fournissent des solutions aux problèmes n'ayant pas de solutions calculables en temps raisonnable de façon analytique ou algorithmique. Les AG sont des approches d'optimisation qui utilisent des techniques dérivés de la science génétique et de l'évolution naturelle : la sélection, la mutation et le croisement et le remplacement.

Il est nécessaire de fournir une fonction permettant de coder une solution sous forme de gènes et une autre fonction pour évaluer la pertinence d'une solution au problème donné. Le principe d'un AG est simple, il s'agit de simuler l'évolution d'une population d'individus jusqu''a un critère d'arrêt.

Un *AG* doit disposer des cinq éléments suivants :

- Principe de codage de l'élément de population : consiste à associer à chaque point de données de l'espace de l'élément, une structure de données.
- Une fonction de génération de la population initiale : ce mécanisme doit être capable de produire une population d'individus non homogènes, qui servira de base pour les générations futures.
- Une fonction à optimiser : une fonction d'utilité (fitness) pour classer les solutions en fonction de leurs aptitudes.
- Des opérateurs : définissent la manière dont les caractéristiques génétiques des parents sont transmises aux descendants (enfants). En effet, l'opérateur de croisement recompose les gènes d'individus existant dans la population et l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'état.
- Les valeurs des paramètres utilisent par l'*AG* : la taille de la population, le nombre total de générations ou critères d'arrêt, la probabilité d'application des opérations de croisement et de mutation.

Un AG doit respecter les critères de convergence suivants :

– Un taux minimum qu'on désire atteindre d'adaptation de la population au problème et un temps de calcul maximum à ne pas d´passer ;
– Une fonction d'utilité à maximiser.

Fonctionnement des algorithmes génétiques

Un algorithme génétique fonctionne de la manière suivante [32] :

Etape 1 : 'Initialisation'

On choisit N individus qui représentent la population initiale (P_0), ils sont tirés aléatoirement.

Etape 2 : 'Evaluation'

On évalue chaque individu par la fonction d'utilité.

Etape 3 : 'Sélection'

On définit les individus de la génération P_i, qui seront dupliqués dans la nouvelle population. A chaque génération, il y a deux opérateurs de sélection : la sélection de reproduction (ou plus simplement la sélection), déterminant les individus qui se reproduiront durant une génération et la sélection pour le remplacement (ou plus simplement le remplacement) qui détermine quels individus devront disparaitre de la population.

Etape 4 : 'Reproduction'

On utilise des opérateurs génétiques (croisement et mutation) pour produire la nouvelle génération. L'opérateur de mutation modifie un individu pour en former un autre tandis que l'opérateur de croisement engendre un ou plusieurs enfants à partir de combinaisons de deux parents.

Etape 5 : 'Retour'

Retour à la phase d'évaluation tant que les conditions d'arrêt du problème ne sont pas atteintes.

Les AG présentent quelques avantages, parmi lesquels on cite [33] :

- Ils autorisent la prise en compte de plusieurs critères simultanément.
- Ils parviennent à trouver de bonnes solutions sur des problèmestrès complexes.
- Ils combinent entre l'exploration de l'espace de recherche et l'exploitation des meilleures solutions disponibles à un moment donné.
- Ils doivent simplement déterminer, entre deux solutions, quelle est la meilleure afin d'opérer leurs sélections.

Toutefois, les AGs ont des limites [33] :

- Le coût d'exécution important : les algorithmes génétiques nécessitent de nombreux calculs, en particulier au niveau de la fonction d'évaluation.
- L'ajustement d'un algorithme génétique est délicat : des paramètres tels que la taille de la population ou le taux de mutation sont parfois difficiles à déterminer. Or, le succès de l'évolution en dépend et plusieurs essais sont donc nécessaires, ce qui limite encore l'efficacité de l'algorithme.
- Lorsqu' une population évolue, il se peut que certains individus, qui a un instant occupent une place importante au sein de cette population, deviennent majoritaires.
- Le caractère indéterministe des algorithmes génétiques : comme les opérateurs génétiques utilisent des facteurs aléatoires, un algorithme génétique peut se comporter différemment pour des paramètres et populations identiques

1.4.4 Approches existants pour la composition des services IOT

Nous allons présenter quelques approches déjà proposées pour la composition de services IoT [16] :

1. **Une approche sociale adaptative basée sur la confiance** pour la composition des services a été proposée [37], qui vise à résoudre la conception et la validation d'un protocole de gestion de confiance adaptatif et survivable pour les systèmes IoT sociaux basés sur SOA ;un utilisateur effectue une évaluation de la confiance sur la base de ses expériences passées de satisfaction directe et des retours de confiance d'autres utilisateurs partageant des intérêts sociaux similaires. les auteurs divisent les relations sociales en trois listes et chaque utilisateur a au moins un appareil haut de gamme désigné (c'est-à-dire un téléphone intelligent et un ordinateur portable) stockant ces listes dans le profil de l'utilisateur. Les autres appareils du même utilisateur ont le privilège d'accéder au profil. en déléguant le stockage et le calcul des réseaux sociaux à un appareil haut de gamme pour chaque utilisateur, de nombreux appareils bas de gamme (c'est à-dire des capteurs) sont capables de partager et d'utiliser les mêmes informations sociales pour maximiser leurs performances. les auteurs ont développé une technique de filtrage adaptatif pour trouver le meilleur moyen de combiner dynamiquement la confiance directe et la rétroaction de confiance indirecte, permettant à chaque nœud de sélectionner de manière adaptative son meilleur paramètre de confiance afin de minimiser le temps de convergence et le biais de confiance.
2. **Une approche probabiliste** a été proposée pour décrire et analyser formellement la fiabilité et les propriétés liées au coût de la composition du service dans l'IoT. les auteurs ont développé une approche pour modéliser la fiabilité et le coût de la composition du service sur la base du MDP[24] avec structure de coût. ainsi, la vérification du modèle probabiliste peut être appliquée pour vérifier et analyser les propriétés de qualité de la composition du service. en utilisant cette approche, la probabilité de réussite du service composite peut être calculée avec les paramètres IoT actuels des appareils. et en outre, la valeur de coût pour chaque service composite candidat peut être obtenue. le développeur

de services IoT peut utiliser les résultats pour prendre la décision concernant la sélection du service et les résultats peuvent l'alerter pour décider s'il doit déployer un service candidat alternatif en cas d'échec du service sélectionné lorsque la probabilité de succès du service sélectionné n'est pas très élevée.

3. **Une approche de composition de services Web RESTful** asynchrone légère pour la composition de services dans l'IoT, qui est basée sur l'extension BPEL[36]. Les auteurs ont divisé l'architecture en six couches et ajouté une interaction asynchrone pour gagner du temps et améliorer les performances.

4. **Un nouvel algorithme de composition basé sur le réseau de Petri** (FindTOptimal)[23] a été proposé pour trouver le chemin le plus optimal, qui utilise une fonction de performance complète nommée (rtc) pour évaluer la rentabilité. pour gérer le changement dynamique des environnements, les auteurs ont proposé un algorithme de surveillance (FbasedMonitor) pour surveiller le système IoT de manière la moins chère.

5. **Une approche middleware** pour la composition des services logistiques dans l'IoT a été proposée. un intergiciel composé d'agents ressources et d'agents tâches est utilisé pour désigner à l'exécution des services composants candidats pour participer à la composition d'une transaction logistique. chaque service composant est représenté par un agent de ressource, qui est chargé de maintenir les informations QoS du service correspondant. Les agents de tâche représentent les instances de service composite et sont chargés de trouver les services composants qui satisfont les exigences de qualité de service de bout en bout de l'utilisateur. les auteurs n'utilisent que quelques moyens de surveillance pour surveiller et assurer les opérations et garantir la robustesse du système logistique[20].

6. **Une Méthode Basé QoS** pour la composition des services a été proposée, les auteurs ont utilisé une prise de décision multi-attributs (MAMD)[35] pour calculer les performances QoS et évaluer chaque service individuel. en fonction des performances QoS et de la valeur fonctionnelle du service, plusieurs services sont choisis pour faire partie de la composition. Les auteurs divisent la composition des services en analyse de la de-

mande des utilisateurs, recherche et correspondance des services, sélection des services et enfin composition des services, et un algorithme génétique amélioré est utilisé pour trouver les solutions optimales de composition des services.

7. **Une cadre pour la composition des services IoT dans l'IoT** a été proposée[18] ; Les auteurs prennent en compte la réputation du prestataire de services et la fiabilité du prestataire de réputation afin d'identifier les meilleurs candidats à la création d'un service composé, pour faire face à une tâche donnée. La réputation d'un fournisseur est calculée localement, de sorte que chaque fournisseur peut calculer une réputation différente en fonction de ses expériences personnelles antérieures avec différents fournisseurs .cette approche est coûteuse en calcul et ne gère pas les échecs de connexion, ce qui n'est pas très adapté aux grands environnements.

le tableau suivant représente une comparaison basée sur les critères suivants :

- Composition dynamique.
- Indépendance et extensibilité.
- L'optimisation.
- La performance.

critères	1	2	3	4	5	6	7
Composition dynamique	+	+	−	+	+	+	+
Indépendance et extensibilité	+	+	+	+	+	+	−
L'optimisation	+	+	−	+	+	+	+
La performance	+	−	+	+	+	+	−

TAB. 1.1 – Tableau de comparaison des approches étudiées

Vis à vis les travaux cités auparavant, nous proposons d'utiliser une méthode automatique, dynamique et optimisée de composition des services IOT basée QoS en utilisant un algorithme évolutionnaire qui permet de choisir le meilleur service composite parmi ceux disponibles.

1.5 Conclusion

Dans ce chapitre, nous avons défini les concepts de base nécessaire et nous allons conclut par analyse de certaines approches existantes, afin d'agrémenter notre solution proposé.

Le chapitre suivant, sera dédié la conception et le développement de notre approche.

Chapitre 2

Une approche de composition de services IOT basée QoS

2.1 Introduction

De nos jours, l'IoT devient de plus en plus répandu et adapté aux interactions des utilisateurs, qui composent les technologies de service de manière hétérogène.

En règle générale, les appareils IoT sont intégrés de manière hétérogène et dynamique avec les fonctions de fiabilité liées à leur QoS pour chaque service atomique. Chaque service atomique est lié au dispositif IoT qui fournit la fonctionnalité du service. Dans la plupart des cas, un seul service n'est pas suffisant pour répondre aux exigences complexes de l'utilisateur.

Dans les chapitres précédents nous avons vu les concepts de base l'internet des objets et les caractéristique et l'architecture de IOT. L'objectif de ce chapitre est de présenter l'essentiel de notre travail qui consiste à développer une approche de composition de services IOT avec QoS, elle est basée sur les algorithmes génétiques dont le but est de trouver une solution optimale (une combinaison de services) qui répond au mieux à la requête de l'utilisateur.

2.2 Motivation

Aujourd'hui le besoin d'un client est devenu de nature multi-objectif (minimiser les couts, minimiser le temps de réponse, maximiser la sécurité, etc.). De plus, étant donné que les services ayant des fonctionnalités similaires soient fournis par des fournisseurs offrant des propriétés non fonctionnelles différentes, il sera donc nécessaire de trouver un outil adapté au contexte afin de choisir le plus adéquat au besoin de l'utilisateur.

2.3 Architecture générale du système

Dans cette partie, nous allons expliquer notre architecture et ses parties, nous avons également présenté le fonctionnement global de cette architecture.

2.3.1 Architecture proposée :

Comme nous l'avons vu dans le premier chapitre l'association de normes IEEE considère trois couches dans l'architecture IoT, comme illustré à la figure 2.1. L'architecture IoT actuelle est trop rigide et pas répondu les besoins complexes de l'utilisateur et il n'est pas construire des dispositifs intelligents.

Notre architecture proposée pour les appareils IoT comporte quatre niveaux. En particulier, nous avons ajouté une couche service de composition dans l'architecture présentée à la figure 1.1, chapitre 1) , Nous avons intégré couche service de composition pour répondre les besoins complexes d'une requête (l'utilisateur ou service).

Nous avons inclus la couche service entre la couche application et la couche réseau ; la raison derrière cela est que la couche service de composition proposé est un mécanisme et qu'il doit utiliser la couche réseau et communication de données pour échanger des informations avec la couche application. La couche service utilise la couche physique (couche perception) pour détecter l'environnement et traiter ces informations réelles.

La figure 2.1 montre une brève description de la composition des services dans l'architecture IoT.

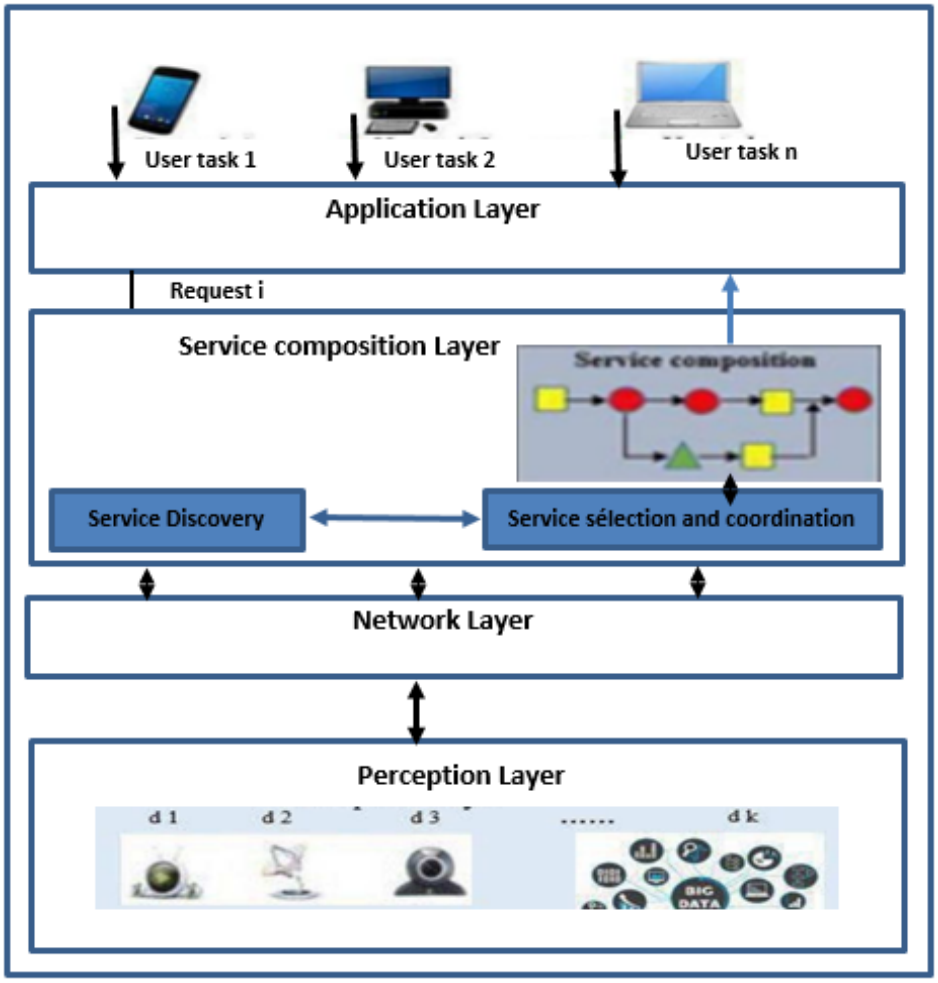

FIG. 2.1 – Architecture proposée[26]

Cette architecture se compose de quatre couches qui sont expliquées comme suit :

1. La couche de perception comprend des capteurs et des appareils intelligents pour collecter les données de l'environnement IoT.
2. La couche réseau permet de se connecter à d'autres serveurs, des objets intelligents tels que des capteurs et des périphériques réseau. L'autre caractéristique importante de cette couche est le transfert des données recueillies à partir de la couche de perception.
3. La couche de composition de service est responsable de la composition d'un certain nombre de sous-services en fonction des exigences fonctionnelles et non fonctionnelles

de l'utilisateur. Cette couche traite de la coordination des demandes livrées pour les envoyer pour découvrir les sous-services disponibles et appropriés en raison des demandes de l'utilisateur enfin les sous-services sélectionnés sont combinés pour composer le service composite souhaité qui doit être livré. à travers la couche d'application.

4. La couche d'application pilote des services composites particuliers pour les utilisateurs finaux en fonction de leur demande.

Dans ce travail, nous nous concentrerons uniquement sur la couche de composition des services et fournirons une étude complète.

2.4 Architecture détaillée

Le but de la composition des services est de créer de nouvelles fonctionnalités en combinant plusieurs services atomiques (ou complexes) fournis par l'environnement IOT afin d'apporter une valeur ajoutée en fonction de la demande des utilisateurs.

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

La composition des services peut être divisée en :

- Analyse de la demande d'utilisateur.
- Recherche et correspondance des services.
- Sélection des services.
- Composition des services.

Le processus de composition des services IoT est illustré à la Figure 2.2 :

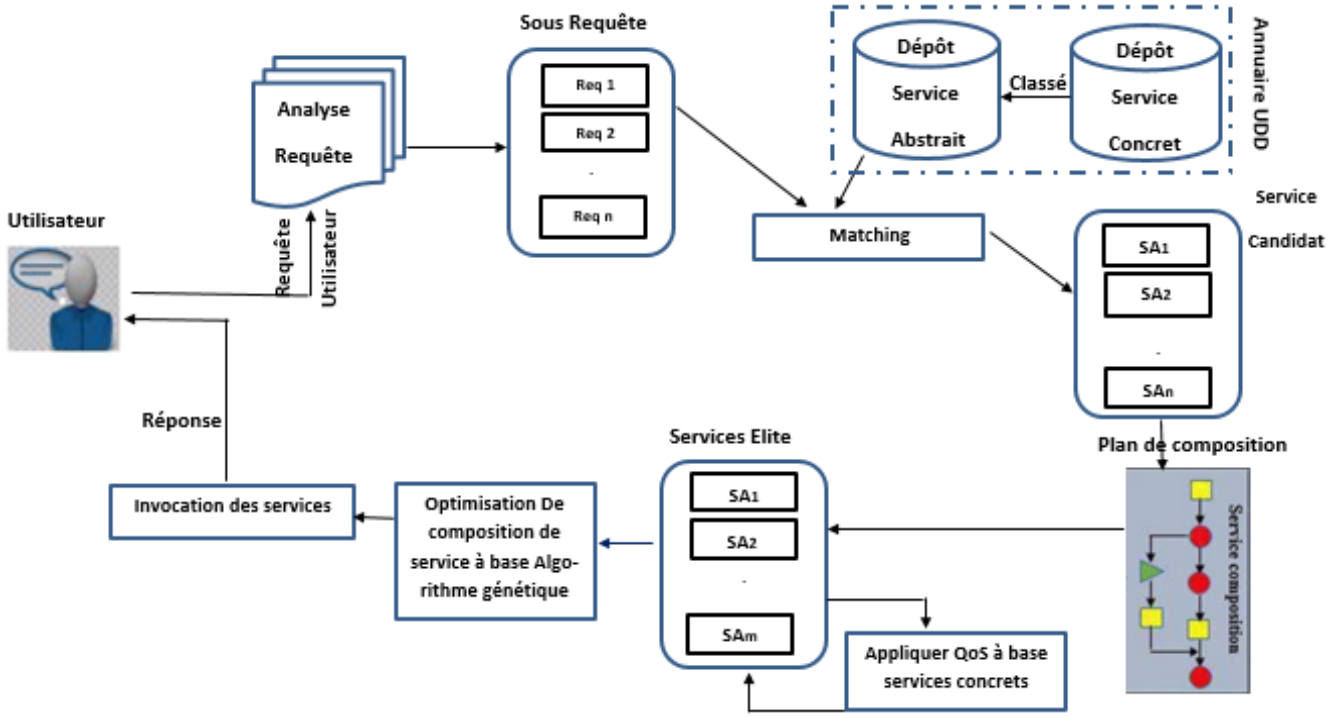

FIG. 2.2 – Modèle de traitement de composition de services IOT

Nous proposons une approche composition de services avec QoS basée sur les algorithmes génétiques qui se déroule selon les étapes suivantes :

Etape 1 : Formulation de la requête

La première étape est faite par l'utilisateur. Celui-ci va formuler sa requête via une interface utilisateur. Dans sa requête, il spécifie les paramètres dont il dispose (les entrées) et ce qu'il recherche (les sorties du système ou le but à atteindre).

Description des services (abstrait et concret) avec structure prédéfinie comme suit :

Entrée 1 , Entrée 2 , ... , Entrée n ; Sortie 1 , Sortie 2 , ... , Sortie n

FIG. 2.3 – Formule de la requête

Etape 2 : La correspondance (Matching)

Quand l'utilisateur introduit sa requête on prend les sorties de cette requête et on compare ces sorties (1 ou plusieurs) avec les sorties des services abstraits si au moins une sortie est similaire on prend ce service et on le met dans une liste service candidat sinon on ne le prend pas.

Algorithm 1 Algorithme de matching

Partie A

Begin

List <String> List_candidat_final ;

List <String> Resulta =" " ;

For tout **SA** du service dépot **do**

For tout sortie service SA **do**

If (sortie requête == sortie service SA) **then**

Resulta += service SA ;

end if

end for

end for

Service_Dependent (Resulta) ;

END

Partie B

Void Service_Dependent (List Resulta) {

String R =" " ;

For tout SA du Resulta **do**

If SA possède Entrée **then**

For chaque Entrée **do**

```
        Candidat = Chercher_SA_sortie_egale_ Entré(Entrée) ;
       If Candidat.length > 0
        List_candidat_final = List_candidat_final + Candidat ;
            For chaque SA dans Candidat do
            List_candidat_final= Service_Dependent (Candidat) ;
             end For
     end If
    end For
 enf If
end For
}
```

Partie C

```
String Chercher_SA_sortie_egale_ Entré(Entrée) {
String Res =" " ;
For tout SA du service dépot do
  If ( SA possède Sortie == Entrée ) then
     Res = Res + SA ;
    end If
 end For
Return(Res) ;
}
```

Etape 3 : Test

Répéter l'étape 2 jusqu'à ce que les points d'arrêt soient aboutis.

Avec les critères d'arrêt sont :

1. le service n'a pas d'entrée.
2. n'y a pas des services à composer.
3. l'arrivé a les entrées de la requête.

Etape 4 : schéma de composition

Généré le plan abstrait globale automatiquement sur les services candidats en utilisant un algorithme de chainage arrière.

– Une fois le plan global est obtenu, un plan abstrait de composition optimal est choisi en fonction de la réputation.
– Cette sélection permet d'éviter l'invocation des services qui fournissent les mêmes paramètres.

Etape 5 : Sélection locale (selon Non fonctionnelle)

1. Calculer le poids associé à chaque paramètre non fonctionnel de chaque service concret.
2. Calculer la somme pondérée (**Qos_cal**) des paramètres non fonctionnels de chaque service concret.
3. Classer et trier les services concrets dans chaque service abstrait selon **Qos_cal** calculé.
4. Tout service n'offrant pas le seuil minimal de qualité de l'utilisateur sera rejeté.

Etape 6 : Sélection globale

L'étape de sélection globale a comme objectif de vérifier la satisfaction des contraintes fonctionnelle afin de sélectionner le meilleur service composite en utilisent l'algorithme génétique.

Etape 7 : Exécution

Appliquer le schéma de composition qui générer dans l'étape 4 pour invoquer les services concrets.

On va expliquer le processus de composition de service par diagramme de séquence suivant :

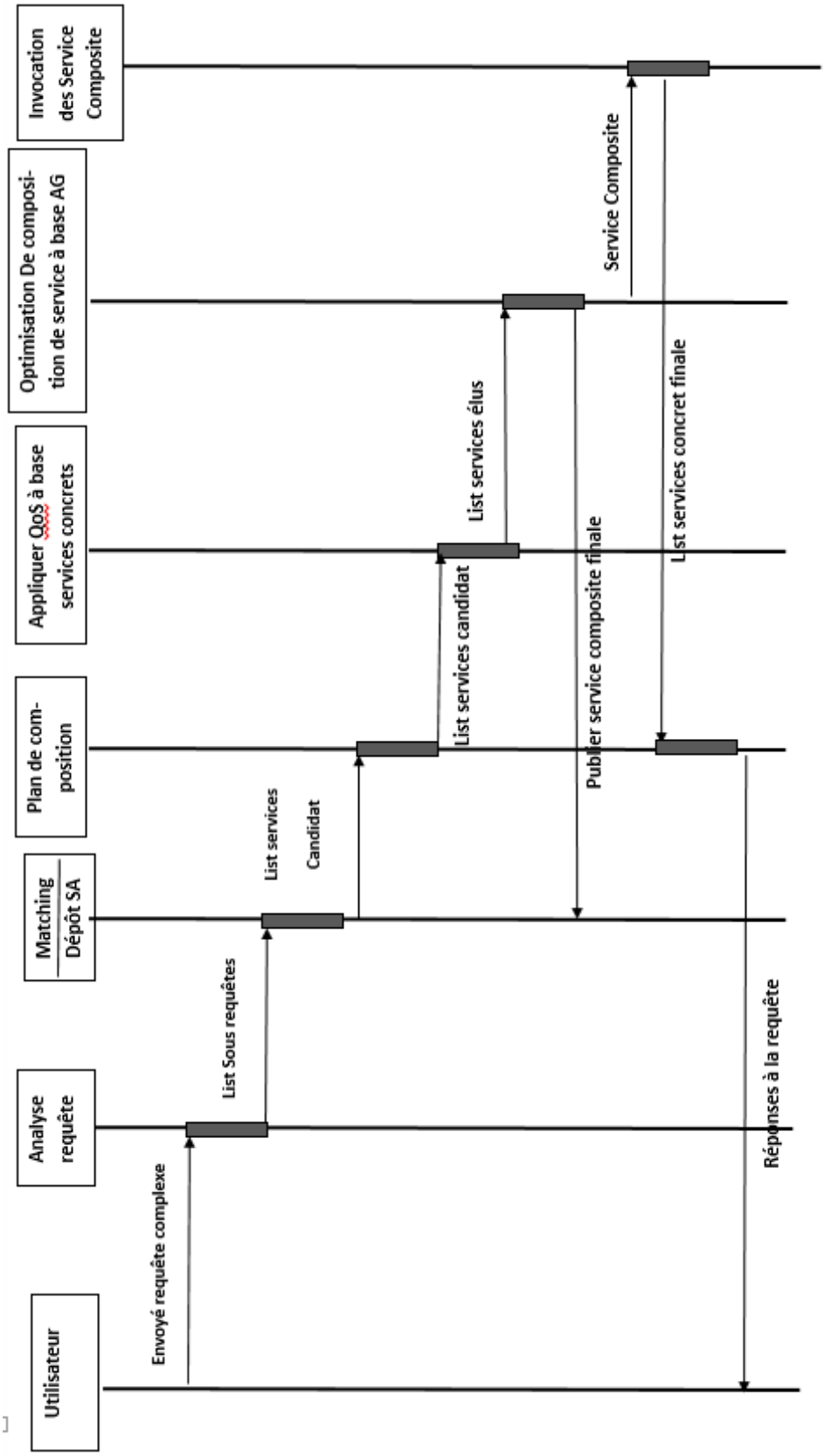

FIG. 2.4 – Processus de composition de service par diagramme de séquence

2.4.1 Modélisation des services

Dans ce travail, on considère les deux types de service : service concret et service abstrait.

2.4.1.1 Service abstrait

Un service abstrait est un ensemble de services concrets, qui ont les mêmes entrées et les mêmes sorties, sans précisions techniques utiles lors de la sélection de services

Un service abstrait est noté : $SA_i =< SA_{in}\ i\ ,\ SA_{out}\ i\ ,SA_{sc}\ i\ ,R_i >$ tels que (voir la figure 2.5) :

SA_i^{in} : ensemble des entrées du service SA_i .

SA_i^{out} : ensemble des sorties du service SA_i .

SA_i^{sc} : ensemble de services concrets du service SA_i tel que : SA_i^{sc} : $\{SC_{i,1}, SC_{i,2}, ..., SC_{i,n}\}$ tel que $\forall SC_{i,j}$et $SC_{i,k} \in SA_i^{CS}$,$(SC_{i,j}^{in} = SC_{i,k}^{in} = SC_i^{in}\ ) \wedge (SC_{i,j}^{out} = SC_{i,k}^{out} = SC_i^{out})$.

R_i : la réputation du service abstrait.

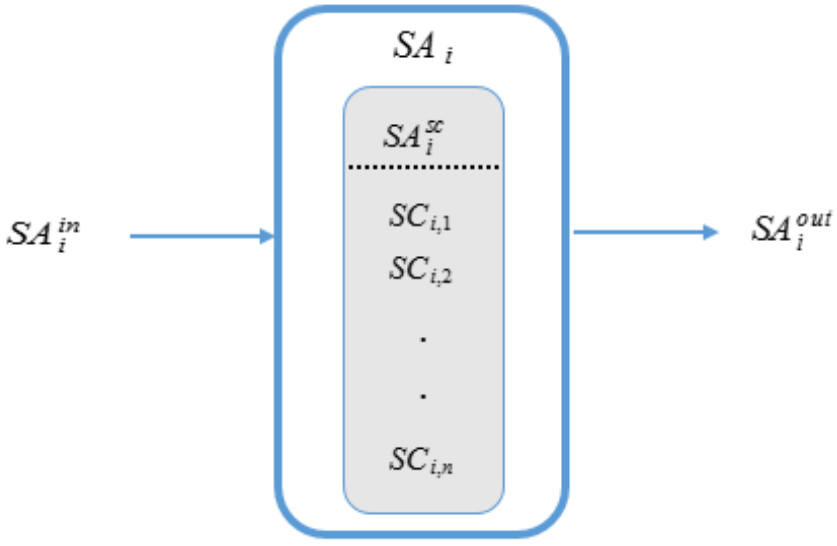

FIG. 2.5 – Représentation d'un service abstrait

2.4.1.2 Service concret

Un service concret (noté SC_i) est une instanciation du service abstrait. Il est défini comme étant une fonction physique ou logique fournissant des données contextuelles et qui peut avoir un effet sur le monde réel [27]. Il prend des données en entrée SC_{in} i , afin de produire des données en sortie SC_{out} i . Un service concret est subdivisé en deux parties la partie fonctionnelle et la partie non fonctionnelle. La partie fonctionnelle est alimentée par des messages d'entrée SC_{in} i et produit des messages de sortie SC_{out} i . La partie non fonctionnelle, quant à elle, représente la QoS du service SC_i (voir la figure 2.6).

Un service concret est décrit par un vecteur $SC_i =< SC_{in}$ i ,SC_{out} i ,$QoS_i >$, tel que :

SC_i^{in} : représente les paramètres d'entrée du service SC_i .

SC_i^{out} : sont les paramètres de sortie du service SC_i .

QoS : désigne les paramètres de QoS représentés sous forme d'un vecteur

$QoS_i = \{q_{i,1}, q_{i,2}, ...,q_{i,n}\}$, ou $q_{i,j}$ représente l'attribut j de QoS du service SC_i

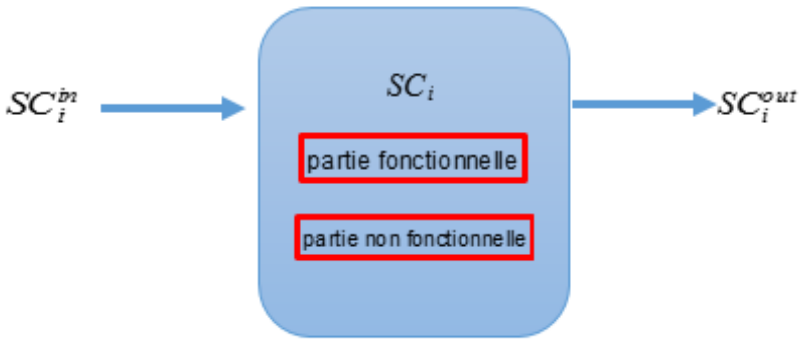

FIG. 2.6 – Représentation d'un service concret.

2.4.2 Qualité de service

La qualité de service (notée QoS) désigne la capacité à fournir un service conforme aux exigences et aux préférences de l'utilisateur (en matière de temps de réponse par exemple). Dans cette section, nous détaillons la manière dont les paramètres de qualité de services sont calculés.

2.4.2.1 Energie

L'énergie représente le niveau de batterie du dispositif IOT qui héberge un service donné. Ce paramétre vise à vérifier si l'exécution du service durera jusqu' à sa fin [28]. Ce paramètre est mesuré en pourcentage. L'énergie est un attribut positif, il est donc à maximiser.

2.4.2.2 Localisation

La localisation permet de satisfaire l'utilisateur en s'adaptant à son contexte d'une manière transparente. L'objectif de prendre en considération cet attribut est de sélectionner le dispositif (service) le plus proche à l'utilisateur pour répondre à sa requête (exemple, pour un service d'affichage sur l'écran, si l'utilisateur est à proximité de son PC la notification sera affiché sur l'écran de ce dernier).

2.4.2.3 Temps de réponse

Le temps de réponse (noté TR) est le temps mis entre l'envoi de la requête par l'utilisateur et la réception du résultat envoyé par le service [29]. Le temps de réponse, appelé aussi latence, représente le temps requis par un service pour répondre à une demande (invocation) [27]. Dans notre travail, le temps de réponse est calculer en se basant sur les exécutions précédentes (voir l'algorithme 1).

Algorithm 2 Algorithme de Calcul du temps de reponse d'un service concret

Begin

TR_{total} :=0 ;

for i :=1 à n **do**

TR_i :=$T_{fin} - T_{debit}$;

TR_{total} :=$TR_{total} - TR_i$;

end for

$TR := TR_{total}/n$;

End

Tel que : TR_{total} est le temps de réponse cumulé d'un service concret SC_i .

TR_{debit} est le temps du début d'invocation du service concret SC_i .

TR_{fin} est le temps de la fin d'invocation du service concret SC_i .

n : est le nombre d'invocations.

2.4.2.4 Fiabilité

La fiabilité (notée $Fiab$) représente l'aptitude d'un service à s'exécuter sans détériorer l'information traitée tout en respectant les demandes et les conditions contractuelles [30]. La fiabilité représente le taux du succès d'un service pour répondre à une invocation (voir l'algorithme 3).

Algorithm 3 Algorithme de Calcul de la fiabilite d'un service concret

Begin

$Fiab_i$:=0 ;

for i :=1 à n **do**

if la requête de l'utilisateur est satisfaite **then**

$Fiab_i := Fiab_i + 1$

end if

end for

$Fiab_i = Fiab_i/n$;

End

Tel que $Fiab$ est la fiabilité de service SC_i.

2.4.2.5 Disponibilité

Ce paramètre prend une valeur entre 0 et 1. En effet, à un instant donné, un service peut être disponible (c.à.d., dispo > 0) ou indisponible (c.à.d., dispo $= 0$). L'algorithme 3 calcule la valeur de ce paramètre.

La disponibilité (notée *Dispo*) est un paramètre imprévisible car il dépend de plusieurs facteurs notamment le niveau d'énergie du dispositif IOT de traitement, la mobilité du dispositif, la disponibilité des serveurs, l'état de la connexion réseau, etc.

Algorithm 4 Algorithme de Calcul de la disponibilite d'un service concret

Begin

NB_{Dispo} :=0 ;

for i :=1 à n **do**

if la requête de l'utilisateur est satisfaite **then**

$NB_{Dispo} := NB_{Dispo} + 1$

end if

end for

$Dispo = NB_{Dispo}/n$;

End

Tel que $Dispo$ est la disponibilité du service SC_i , et NB_{Dispo} est le nombre de fois où le service est disponible.

2.4.2.6 Prix

Cet attribut représente le coût monétaire que l'utilisateur doit payer pour utiliser une ressource, il est obtenu en compensant le coût du matériel et des logiciels. La plupart des fournisseurs fixent des prix différents pour leurs services.

2.4.2.7 Réputation

Le terme réputation exprime généralement l'opinion du publique au sujet de quelque chose ou de quelqu'un (service, personne, un groupe de personnes, une organisation, une ressource, etc.). Par analogie, la réputation d'un neoud dans un réseau représente l'opinion des neouds de ce réseau vers la disponibilité et la fiabilité de ce dernier.

D'une manière plus simpliée, la réputation exprime le degré de la confiance attribuée à un service IOT par l'ensemble ou une partie de l'ensemble des services du system IOT[31].

Calculer la Réputation de chaque service abstrait

La réputation d'un service abstrait (noté *Réputa*) exprime l'opinion de l'ensemble des poids à chaque critère de QoS de services concret.

$\boldsymbol{W(Energie) = 0.4;}$

$\boldsymbol{W(Fiab) = 0.6;}$

Algorithm 5 Algorithme de Calcul de la reputation d'un service Abstrait

Begin

Double Répu, Réputa ;

for tout SA **do**

for tout $SC_i \in SA$ **do**

Répu $SC_i = \sum$ (0,4 * $\boldsymbol{Enrgie}$) + (0,6 * $\boldsymbol{Fiab}$) ;

end for

Réputa = Répu/n ; // n nombre service concret

end for

End

2.4.3 Mise à l'échelle de qualité de service

Les attributs de QoS sont divisés en deux classes, les attributs positifs et les attributs négatifs. Les valeurs des attributs positifs doivent être maximisées (disponibilité, fiabilité et sécurité). En revanche, les valeurs des attributs négatifs doivent être minimisées (coût et temps de réponse).

Pour faire face à ce problème, une phase de mise à l'échelle doit être effectuée pour normaliser les valeurs des attributs de QoS.

La mise à l'échelle est une phase de prétraitement visant à normaliser les valeurs de QoS

associées à des attributs positifs et négatifs, en les transformant en une valeur comprise entre 0 et 1.

La QoS d'un service candidat SC_i est représentée sous forme d'un vecteur $QoS(SC_i) =< q_{i,1}, q_{i,2}, ..., q_{i,n} >$,

où n représente le nombre d'attributs de QoS requis par l'utilisateur

et $q_{(i,j)}$ représente la valeur de l'attribut $j(1 \leqslant j \leqslant n)$ de QoS du service SC_i .

Afin de normaliser les valeurs des attributs de QoS, on utilise les formules suivantes :

– **Attributs négatifs :**

$$q'_{(i,j)} = \begin{cases} \frac{q_j^{\max} - q_{(i,j)}}{q_j^{\max} - q_j^{\min}} & \text{si } q_j^{\max} - q_j^{\min} \neq 0 \\ 1 & \text{sinon} \end{cases}$$

– **Attributs positifs :**

$$q'_{(i,j)} = \begin{cases} \frac{q_{(i,j)} - q_j^{\min}}{q_j^{\max} - q_j^{\min}} & \text{si } q_j^{\max} - q_j^{\min} \neq 0 \\ 1 & \text{sinon} \end{cases}$$

$q'_{(i,j)}$ dénote la valeur normalisée de l'attribut $j$de QoS associé au service concret SC_i , elle est calculée en utilisant la valeur actuelle $q_{i,j}$ ainsi que les valeurs maximales et minimales de l'attribut j de QoS.

QoS_i qui se réfèrent respectivement aux $q_j^{\max}$ et $q_j^{\min}$

On suppose que les domaines de valeurs de critères de QoS sont définit comme suit :

critères	**domaines de valeurs**
Energie	0. 100
Localisation	0. 100
Temps de réponse	1. . . . . 300 Ms
Fiabilité	0. 100
Disponibilité	0. 100
Prix	0. 100

TAB. 2.1 – les domaines de valeurs de critères de QoS

Affectation de poids

On peut exprimer les mesures de service en termes de QoS en affectant un poids à chaque critère de QoS. Puis, le courtier de QoS calcule le score de chaque service qui représente une somme pondérée des valeurs des critères de QoS normalisées.

$\boldsymbol{W(TR) = 0.3;}$

$\boldsymbol{W(Fiab) = 0.2;}$

$\boldsymbol{W(Prix) = 0.05;}$

$\boldsymbol{W(Energie) = 0.3;}$

$\boldsymbol{W(Local) = 0.1;}$

$\boldsymbol{W(Dispo) = 0.05;}$

2.5 Processus de composition de service IOT

2.5.1 Phase de sélection locale

La sélection locale consiste à élire tous les services concrets dont la qualité dépasse le seuil (QoS_locale) spécifié par l'utilisateur. En d'autre terme, QoS est une somme pondérée des paramétrés de qualité de service. Le poids associé à chaque paramètre, est mesuré en fonction des préférences de l'utilisateur (spécifié dans son profil), son contexte et le contexte de

l'environnement. Ce poids reflète le degré d'importance d'un attribut donné pour l'utilisateur.

Algorithm 6 Algorithme de Sélection locale des services concrets

Begin

for tout SA du plan optimal **do**

for tout $SC_i \in SA$ **do**

$QoS_i = \sum (\mathrm{w}_i * \boldsymbol{q}_i)$;

if $QoS_i > QoS_{local}$ **then**

Service SC_i élu ;

else

Service SC_i rejeté ;

end if

end for

end for

End

2.5.2 Phase de sélection globale

Le plan obtenu de la phase précédente (sélection locale) répond aux exigences de l'utilisateur localement. L'objectif de sélection globale à vérifier la satisfaction des contraintes globales afin de sélectionner le meilleur service composite. En effet, lorsque les services concrets formant ce plan sont combinés ensemble, le service composite obtenu doit satisfaire la contrainte de la qualité globale (c à d, maximiser la fonction d'utilité).

2.5.2.1 Fonction d'agrégation de QoS

Selon le plan (schéma) de composition, nous associons à chaque Attribut de QoS considéré une valeur :

Attribut de QoS	**Séquentielle**	**Parallèle (et)**
Energie	$\prod_{j=1}^{n} Energ_j$	$\prod_{j=1}^{n} Energ_j$
Localisation	$\sum_{j=1}^{n} Local_j$	$\sum_{j=1}^{n} Local_j$
Temps de réponse	$\sum_{j=1}^{n} TR_j$	$MAX_{j=1}^{n} TR_j$
Fiabilité	$\prod_{j=1}^{n} Fiab_j$	$\prod_{j=1}^{n} Fiab_j$
Disponibilité	$\prod_{j=1}^{n} disop_j$	$\prod_{j=1}^{n} disop_j$
Prix	$\sum_{j=1}^{n} prix_j$	$\sum_{j=1}^{n} prix_j$

TAB. 2.2 – Fonction d'agrégation de QoS

- **Séquentielle** : une séquence de services $\{S_1, ..., S_n\}$ est exécutée dans un ordre chronologique strict.
- **Parallèle (ET)** : plusieurs services $\{S_1, ..., S_n\}$ sont exécutés en même temps.

2.5.2.2 Les paramètres de l'AG

La solution quasi-optimale (représenté par un chromosome) est déterminé par l'évolution d'une population initiale à travers un nombre déterminé de générations en appliquant des

opérateurs génétiques jusqu'à ce qu'un critère d'arrêt soit satisfait .Dans ce qui suit nous détaillons tous ces paramètres.

Le codage

Les éléments de la population sont codés de la manière suivante :

- **Le chromosome** : représente un service composite qui est une chaine de gènes représentant les solutions potentielles.
- **Gène** : représente un service concret.
- **La population** : un ensemble de composition (caractérisé par des chromosomes) où les services composants sont sélectionnés aléatoirement.
- **La fonction d'utilité** : elle joue un rôle prépondérant dans le déroulement de l'algorithme génétique, elle sert à maximiser la valeur de qualité globale ($QoS_{globale}$) du service composite.

L'objectif ici est de trouver un service composite $S_{composite} = \{SC_1, SC_2, ..., SC_n\}$ tel que :

- Les valeurs des attributs de QoS de $S_{composite}$ sont maximisées ;
- Les contraintes globales de l'utilisateur sont satisfaites ;
- Le nombre maximale de générations n'est pas atteint.

Création de la population initiale

Dans cette étape, la taille de population initiale est générée par l'utilisateur en sélectionnant N services composites parmi toutes les combinaisons possibles, ce nombre reste le même pour toutes les populations qui seront générées.

Evaluation

Dans cette étape, on évalue chaque service composite $S_{composite}$ en calculant sa fonction d'utilité (notée F) .

La fonction d'utilité F d'un service composite $S_{composite}$ est calculée en utilisant la formule suivante :

$$F_i = \sum_{j=1}^{nb\_qos} w_j * Q'_j$$

nb_qos est le nombre de paramètre de QoS ;

w_j est le poids du j^{eme} attribut de QoS ;

Q'_j est la valeur agrégée du j^{eme} attribut de QoS ;

Algorithm 7 Algorithme d' évaluation de la population t

Begin

Double F, $F\_optimale\_pop = -1.0$;

for tout $S_{composite}$ de la population **do**

$F_i = \sum_{j=1}^{nb\_qos} w_j * Q'_j$;

if $(F > F\_optimale\_pop)$ **then**

$F\_optimale\_pop = F$;

end if

end for

Return $(F\_optimale\_pop)$;

END

Critères d'arrêt

L'algorithme génétique s'arrête dans l'un des cas suivants :

1. Atteindre le nombre maximal de générations (nbr_{gen_max}) ;
2. Le seuil globale ($seuil\_fitness\_composite$) spécifié par l'utilisateur est satisfaites.

2.5.2.3 Description des opérateurs génétiques

Mutation

En sélectionnement un service (parent) qui a la fonction d'utilité F est maximale, la mutation consiste à changer, dans le service parent ($S_{composite}$), le service concret SC_i possédant la QoS_i minimale (noté $SC_{\min}$) par un autre service (noté $SC_{\max}$) ayant une meilleure QoS_i et appartenant au même service abstrait.

Croisement

En sélectionnement deux services composites (parent 1 et parent 2) ayant la parent 1 possédant la meilleur fitness F . parent 2 sera sélectionné aléatoirement.le croisement consiste en une opération entre deux service composites.

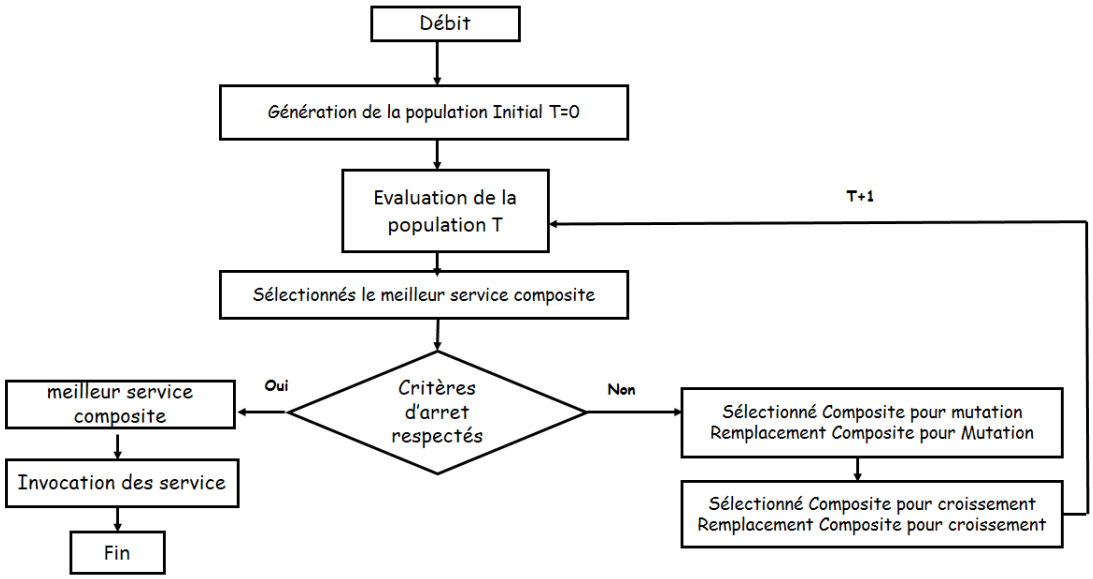

FIG. 2.7 – Organigramme de sélection globale de services

2.5.3 Phase de déploiement et exécution des services

Invoquer les services concrets pour l'exécution en utilisant le schéma de composition optimale selon l'ordre établi.

2.5.4 Phase de contrôle du plan d'exécution

En raison des changements dynamiques de l'environnement, des ressources et de l'évolution des besoins des utilisateurs et du contexte, le plan de composition doit être souple et tolérant

aux pannes. A cet effet, l'approche proposée intègre un mécanisme d'apprentissage sur la QoS du service concret et de la réputation du service abstrait. Tous les paramètres de qualité de tous les services concrets et la réputation de tout service abstrait sont également initialisés Selon le résultat du service invoqué, (succès ou échec) le mécanisme d'apprentissage met à jour le service concret en calculant ses nouveaux paramètres de qualité et estime la nouvelle réputation de son service abstrait.

2.6 Scénario d'application

2.6.1 IOT dans le secteur agricole

L'agriculture est un domaine qui s'est complexifié au fil des années en raison de la croissance démographique (qui augmente la demande de nourriture) et des conditions météorologiques de plus en plus extrêmes et imprévisibles. Grâce à l'IOT, les agriculteurs peuvent désormais obtenir des données et analyses détaillées, notamment pour suivre les conditions de croissance, et mettre ainsi toutes les chances de leur côté pour réussir leurs récoltes.

Les exploitations agricoles étant le plus souvent composées de grands espaces ouverts, où de nombreuses machines, généralement très coûteuses, sont dispersées sur plusieurs hectares. La possibilité de surveiller le bétail, les machines et autres équipements permet donc aux agriculteurs de protéger leurs moyens de subsistance. Par ailleurs les dispositifs et applications IOT peuvent aider les agriculteurs à prendre de meilleures décisions en étant mieux informés pendant la période de croissance.

Par exemple, les appareils intelligents peuvent fournir des informations importantes sur l'état du sol ou de la culture. Les notifications automatiques permettent aux agriculteurs de garder une longueur d'avance et d'anticiper d'éventuels problèmes. Les relevés sur l'état du sol leur permettent également de faire des choix plus écologiques quant à l'utilisation d'eau et d'engrais [34].

Afin d'illustrer le déroulement de l'approche proposée nous utilisons le scénario représentant un serres agricole intelligent. Il s'agit particulièrement de surveiller le mouvement des données capturer dans cet serres et de transmettre ensuite les informations nécessaires aux fournisseurs de service.

Une serre agricole est choisi comme un environnement disposant d'un certain nombre de dispositifs IOT pour offrir des services aux système agricole intelligent, parmi lesquels on cite

- **Des dispositifs d'affichage** : téléphone portable de type smartphone, Tablet, moniteur de PC,etc.
- **Caméras de surveillance** : elle se charge de prendre des photos des plantes pour spécifier le stade phénologique.
- **Des dispositifs de traitement :** pour traiter et analyser les données .

2.6.2 Capteurs utilisés dans le scénario

Le capteur est un dispositif transformant l'état d'une grandeur physique et ou logique observée en une grandeur utilisable. Dans le tableau ci-dessous on cite les capteurs utilisés dans l'environnement :

Capteur	**Description**
Capteur d'évaporation	Calculer le taux d'évaporation
Capteur de transpiration	Connaissant le volume d'eau transpirée
Capteur de température de l'air	déterminer la température de l'air
Capteur de température de sol	déterminer la température de sol
Capteur d'humidité de sol	déterminer d'humidité de sol
Capteur CO_2	Capteur donnat une information sur le taux de CO_2
Capteur Oxygène	mesure la quantité d'oxygène O_2
capteur de pression (tensiomètre)	Calculer le point de flétrissement

TAB. 2.3 – description des capteurs IOT existants dans la serre agricole

2.6.3 Actionneurs utilisés dans le scénario

L'actionneur convertit l'énergie électrique en mouvement mécanique.

Actionneurs	Description
vanne d'irrigation	planifiez et programmez vos l'irrigation
vanne de fertialisation	Station de fertilisation automatique

TAB. 2.4 – description des Actionneur IOT existants dans le serres agricole

2.6.4 Description du scénario

Notre scénario consiste en une ferme avec un groupe de serres intelligentes, dans laquelle l'agriculteur Omar se rend chaque matin afin de mener à bien ses activités quotidiennes, en inspectant les appareils intelligents situés à l'entrée de chaque serre qui déterminent ses tâches quotidiennes,

Omar a cliqué sur la tâche d'irrigation dans le tableau intelligent afin de connaître l'heure et la durée de l'irrigation et s'il a ajouté ou non la fertilisation à l'arrosage.

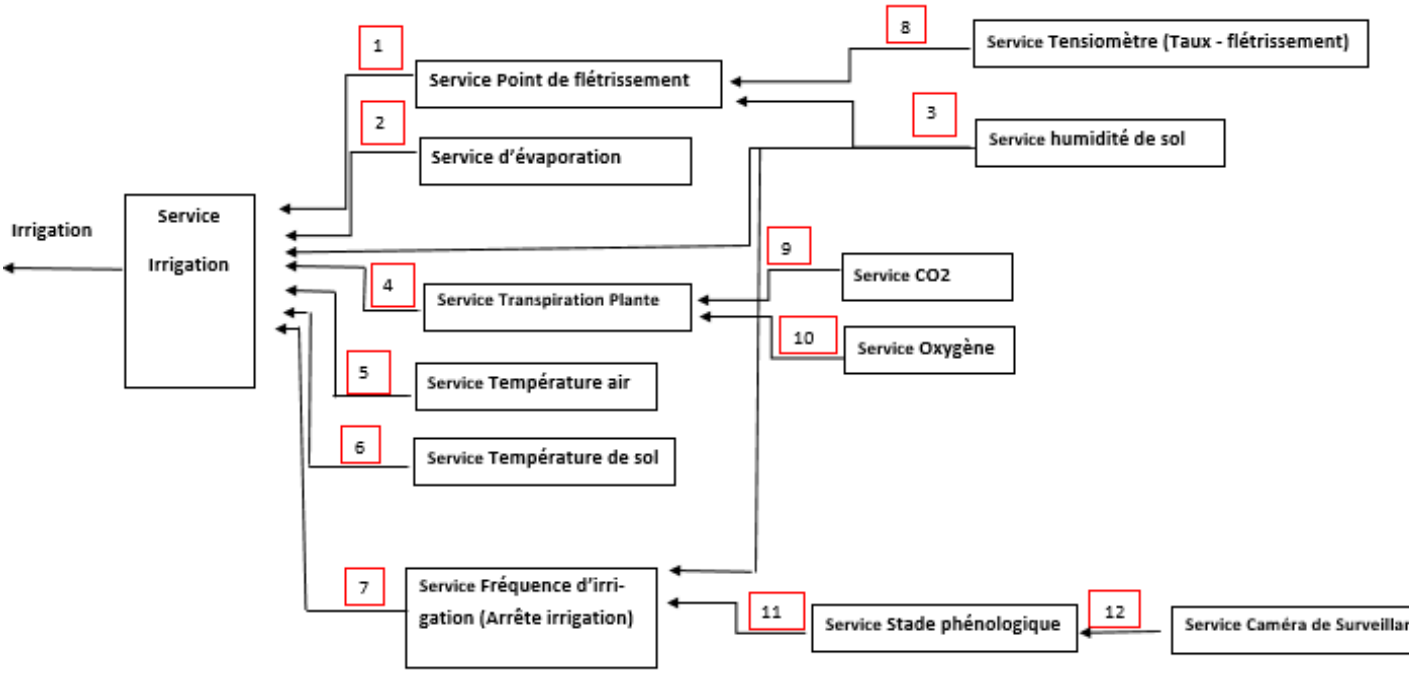

FIG. 2.8 – Scénario applicatif irrigation : les services abstraits

Dans le tableau suivant on donne description des sorties des services abstraits :

N : ° Sortie	**Description**
1	Point flétrissement réversible ou irréversible
2	Taux d'évaporation
3	Taux humidité de sol
4	Taux Transpiration Plante
5	Température air
6	Taux Température de sol
7	Taux Fréquence d'irrigation (Arrête Irrigation)
8	Tensiomètre (Taux - flétrissement)
9	Taux CO2
10	Taux Oxygène
11	Taux de Stade phénologique

TAB. 2.5 – description des sorties des services abstraits "irrigation"

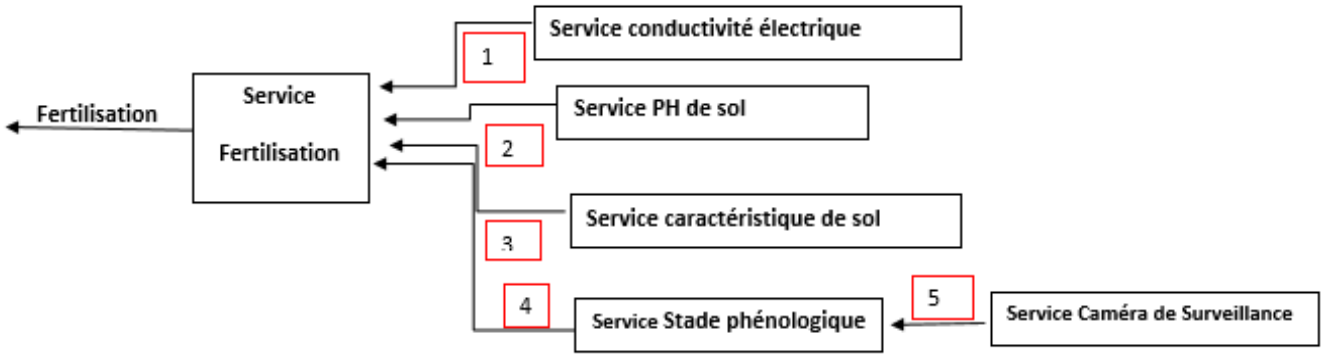

FIG. 2.9 – Scénario applicatif Fertilisation : les services abstraits

Dans le tableau suivant on donne description des sorties des services abstraits :

N : ° Sortie	**Description**
1	Taux conductivité électrique
2	Taux PH de sol
3	caractéristique de sol
4	Taux de Stade phénologique
5	Photo des plantes

TAB. 2.6 – description des sorties des services abstraits "Fertilisation"

2.7 Conclusion

Dans ce chapitre, nous avons proposé une approche de composition de services IOT basé QoS, Ensuite nous avons expliqué les différentes étapes de la composition de service et comment introduire les algorithmes génétiques à la composition de service afin d'obtenir le meilleure service composite qui répond aux exigences locales et globales de l'utilisateur.

Dans le chapitre suivant nous allons expliquer la partie expérimentation et résultats de notre travail.

Chapitre 3

Implémentation et résultats

3.1 Introduction

Dans ce chapitre, nous montrons les étapes d'implémentation et les différentes captures d'écran de notre projet.

3.2 langage et les outils de développements :

La validation de l'approche proposée a été réalisée sur une machine dont les caractéristiques sont :

- Processeur : Intel Core(TM) 2 Duo, avec une vitesse de 2.00 GHZ,
- Capacité de la RAM : 4 Go,
- Système d'exploitation : Windows 8.1 professionnel.

3.2.1 Le langage JAVA

Pour le choix de programmation de notre système nous avons opté pour le langage JAVA et cela pour de nombreuses raisons :

1. JAVA est un langage orienté objet simple, qui réduit le risque des erreurs d'incohérences.

2. Il est indépendant de toute plateforme, il est possible d'exécuter des programmes JAVA sur tous les environnements qui possèdent une Java Virtual Machine (JVM).
3. Il est doté d'une riche bibliothèque de classes, comprenant la gestion des interfaces graphiques (fenêtres, menus, graphismes, boites de dialogue, contrôles), la programmation multi-threads (multitâches) et la gestion des exceptions.
4. Il permet un accès aux bases de données simplifié soit à travers la passerelle JDBC-ODBC ou à travers un pilote JDBC spécifique au SGBD.
5. Il est caractérisé aussi par la réutilisation de son code ainsi que la simplicité de sa mise en œuvre.

3.2.2 Netbeans

Netbeans possède de nombreux points forts qui sont à l'origine de son énorme succès dont les principaux sont :

– Une plateforme ouverte pour le développement d'applications et extensible grace à un mécanisme de plug-in ;
– La construction incrémentale des projets Java grâce à son propre compilateur qui permet en plus de compiler le code même avec des erreurs, de générer des messages d'erreurs personnalisés, de sélectionner la cible.
– Support de plusieurs plates-formes d'exécution : Windows, Linux, Mac OS, etc.
– Un historique local des dernières modifications.

3.2.3 MySQL

MySQL est un serveur de bases de données relationnelles Open Source.

Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de

l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL"

signifie "Structured Query Language" : le langage standard pour les traitements de bases de données.

3.3 Description du système

3.3.1 Structuration de l'algorithme de composition

Nous avons utilisé le langage JAVA pour implémenter notre algorithme de composition de services Le code source englobe un ensemble de classes, chacune implémente une étape de l'algorithme.

- **La classe Principale Service_Composite_IOT_main :** Classe contient la fenêtre principale.
- **La classe Matching :** Quand l'utilisateur introduit sa requête on prend les sorties de cette requête et on compare ces sorties (1 ou plusieurs) avec les sorties des services abstrait si au moins un paramètre est similaire on prend cette service et on le met dans une liste service candidat sinon on ne le prend pas.
- **La classe Plan_De_Composition :** Généré le plan abstrait globale automatiquement sur les services candidat en utilisant un algorithme de chainage arrière.
- **La classe Générer_Service_Concret :** Générer plusieurs services concrets dans chaque service abstrait automatiquement.
- **La classe Algo_G_Agricole :** Il contient les méthodes de mise en œuvre des opérations génétiques (mutation, croisement, etc.) pour la sélection du meilleur service composite.
- **La classe Calculer_QoS_Service_Concret :** Calculé la somme pondérée (Qos_cal) des paramétres non fonctionnels de chaque service concret.
- **La classe Classé_Service_Concret :** permet de classer les services concret qui ont les même entées et les même sorties dans le même service abstrait.
- **La classe Connexion :** Faire la connexion à la base de données.
- **La classe Courbe_AG_Fitness :** Construire et afficher la courbe des valeurs fitness

optimale de chaque génération dans le processus de composition de service.

- **La classe Invocation_Service_Composite_optimale :** Invoquée les services concrets et modifier les attributs de QoS de chaque service, recalculé Qos locale et réputation de chaque service abstrait.
- **La classe Sauvegarder_Scénario :** permet de sauvegarder tous les informations de processus de composition à chaque requête exécuter .

3.3.1.1 Extrait de code les classe principale

```
public class Matching {
    String SA_sor = "";
    Integer indic=0;
  String[] niveau_execut ;
  String niv_exc="";
  int e=0;
    String rout ="";
    Boolean boolea;
    Boolean boool;
    String analys_requet(String re){...37 lines }

String  supprimer_doublon(String service_candidat)
        {...13 lines }
   String  supprimer_doublon_combo(String service_candidat)
        {...13 lines }
String service_dependent(String res,Integer niveau) {...60 lines }
    private String chercher_SA_sortir_egal_entre(String sor) {...49 lines }

    private String applique_reputation(String service_candidat) {...8 lines }

private Boolean tester_bouclage(String resulta, String g) {...15 lines }

    private String tester_entree_service(String resulta) {...39 lines }

}
```

FIG. 3.1 – Extrait de code : classe Macthing

.

```
package service_composite_iot;

import ...8 lines

/**...4 lines */
public class Plan_De_Composition {
 String[][] mat;
//String fileName = "my-file.txt";
 String list_neouds,niv_SA; String[][] niveau_service_abstrait;
 public Integer long_matric;
 Integer long_mat; String SA_sortie; int l,niveu;
 public int [] neoud;
    String [][]  chainage_arriere(String lst) {...45 lines }
    String niveau_chaque_service_candidat(String niv_e)
    {...45 lines }
    String[][] niveau_chaque_service_candidat_mat(String niv_e)
    {...45 lines }
    public void afficher_Matrice()
    {...14 lines }

  void chainag_arrier_matric(int neoud[]) {...66 lines }
    private String chercher_SA_sortir_egal_entreeser(String sor) {...29 lines }

    public void remplissage_Matrice(int l,String resulta,String sor,Integer niveau)

    void plan_Optimale(String niv) {...7 lines }
}
```

FIG. 3.2 – Extrait de code : classe plan de composition

.

```
String Operation_Génétique(String niv_e,String tail_p,String nb_itr,String gra,String reqt,Instant instantStarted)
{    tail_pop =Integer.parseInt(tail_p);
     nombre_generation =Integer.parseInt(nb_itr);
      System.out.println("service abstrait   "+niv_e);
     Intialiser_Population(niv_e);
   while (!Evaluer_Pop_Condition_Arret(generation, composite_optimale))  // cal fitness = eval
     {
         // System.out.println("debit des operation genetique");
         Mutation();
         Croisement();
         String composite_optimal_tout_popuation= Cal_fitness_Fitness_Globale();
         affich_matrice_concret();
         return_matrice_concret();

         Service_Composite_IOT_main  bn =new Service_Composite_IOT_main();
         bn.affiche_mat_concret_jtxt(matrice_pop_concret);
         System.out.println("generation num :   "+generation);
         generation++;

   }
```

FIG. 3.3 – Extrait de code : classe Algorithme génétique

.

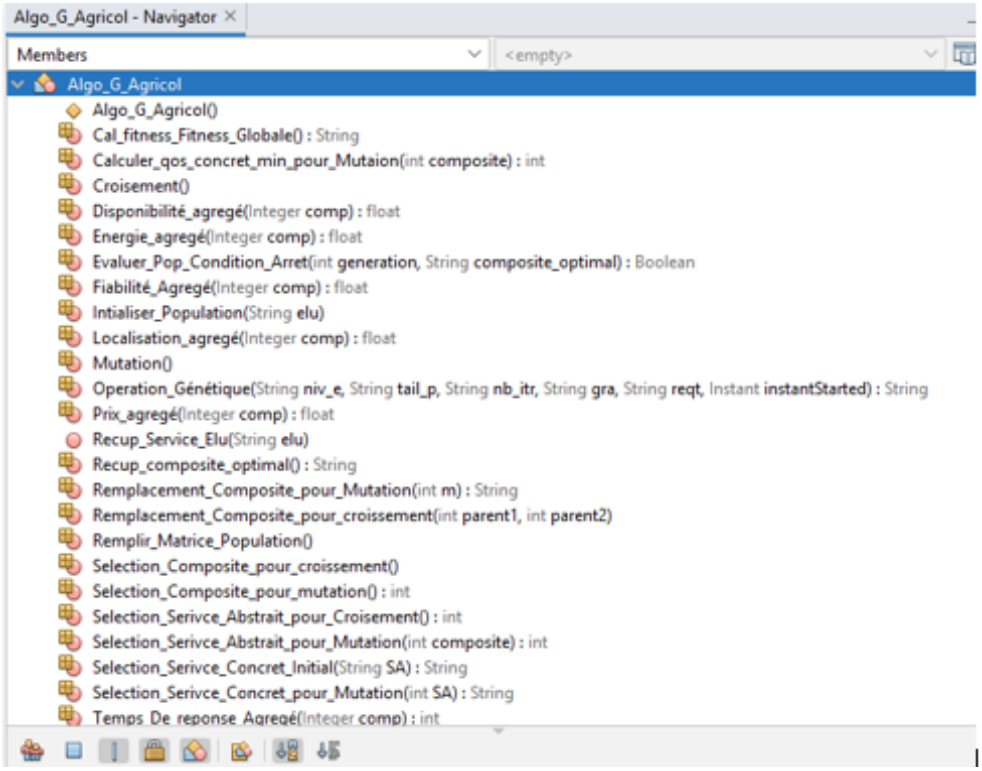

FIG. 3.4 – Méthodes de L'algorithme génétique

3.3.2 Présentation de l'application

Dans notre application, nous distinguons une fenêtre qui présente notre travail.

En cliquant sur le bouton **Actualiser** dans le figure 3.5 pour :

- établir la connexion avec la base de données MYSQL ;
- afficher les services concrets.
- classer les services concrets dans dépot services abstrait et afficher.
- calculer QoS de chaque service concret.
- calculer réputaion de chaque service abstrait.
- charger tous les informations à partir la base de données dans l'interface utilisateur.

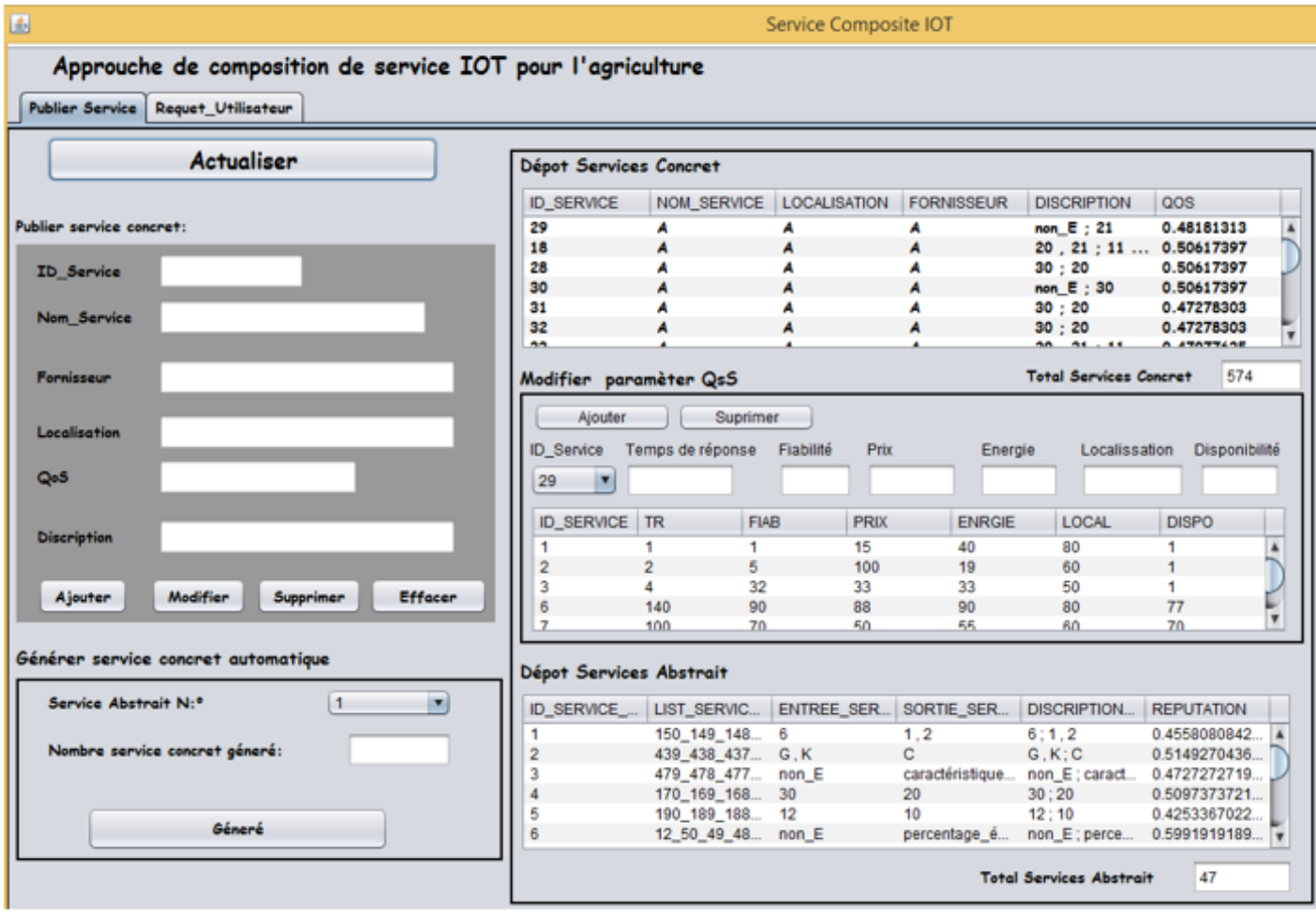

FIG. 3.5 – fenêtre principale de projet

..

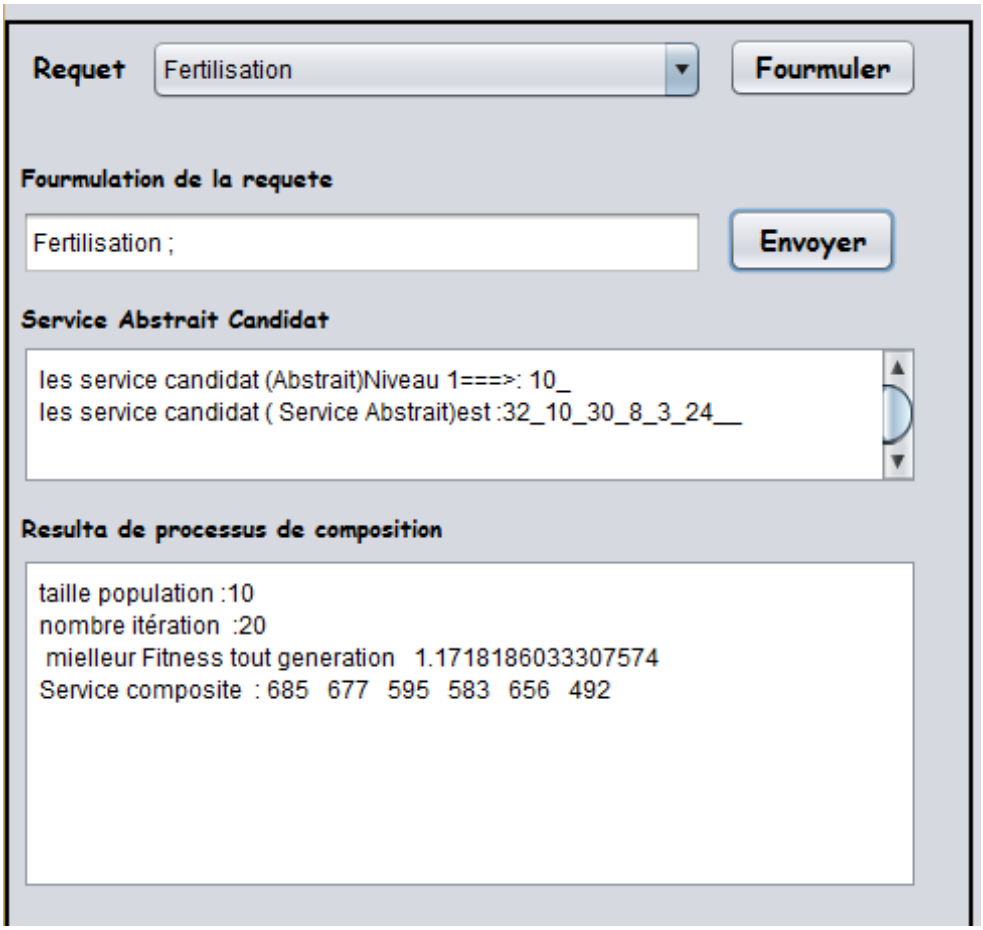

FIG. 3.6 – Formulation de requet utilisateur

La figure 3.6, l'utilisateur formule la demande ou plusieurs et envoi la requête pour obtenir la réponse..

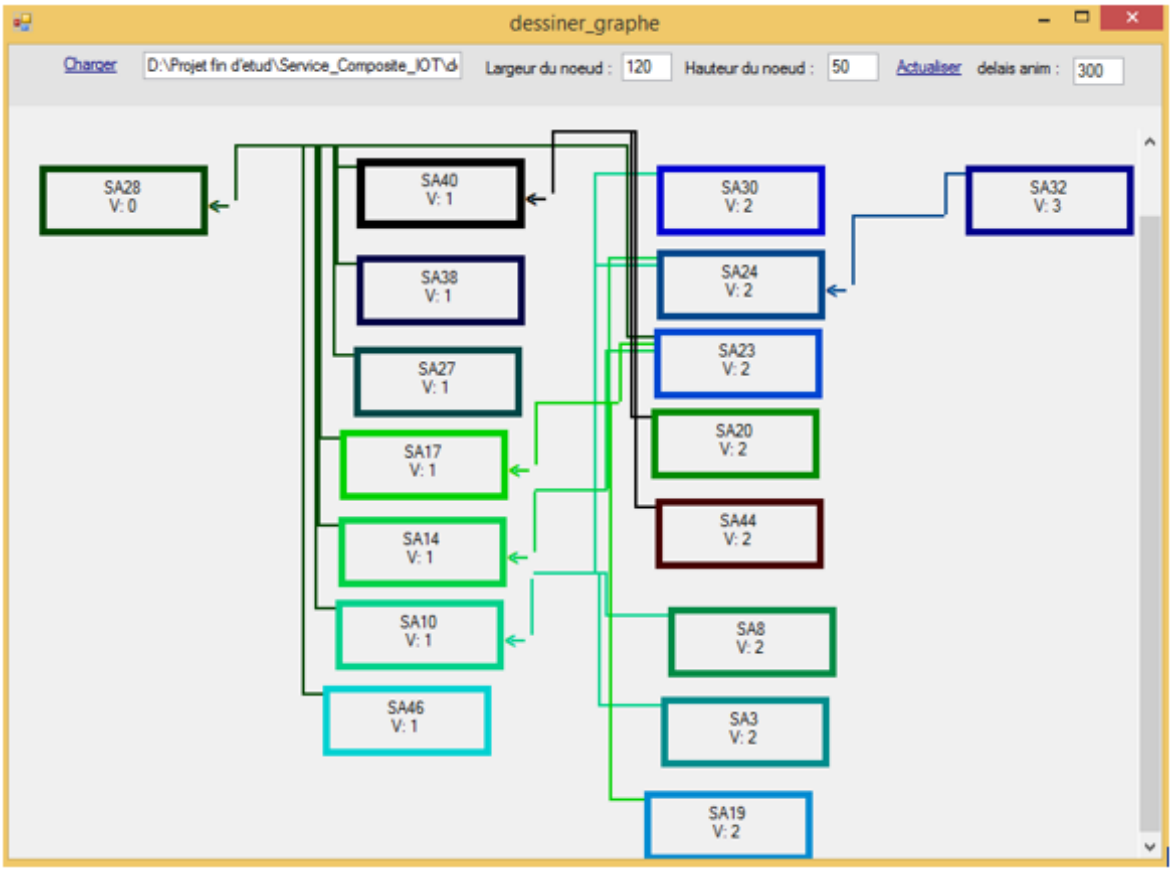

FIG. 3.7 – Plan de Composition des services abstrait " Irrigation "

.

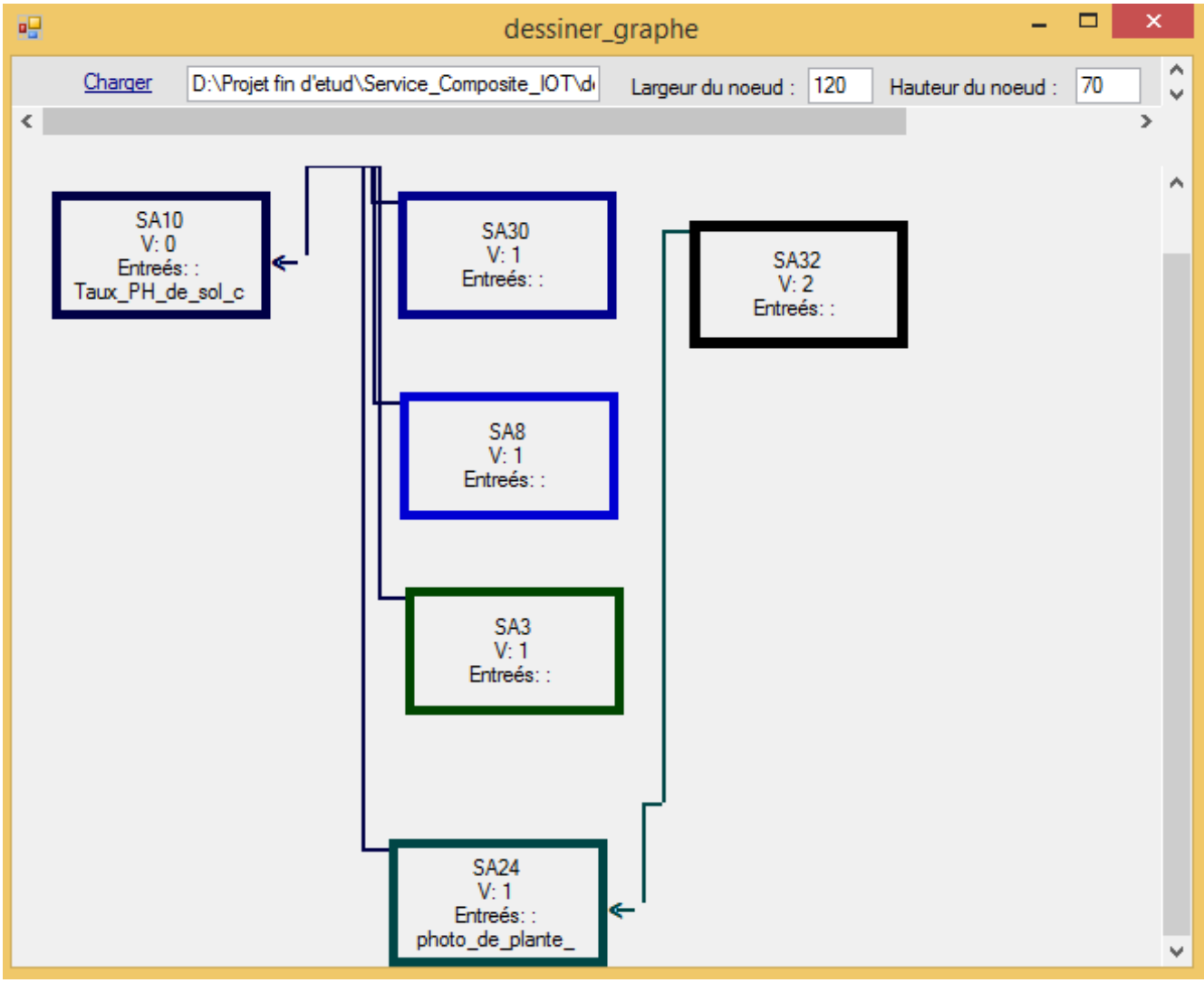

FIG. 3.8 – Plan de Composition des services abstrait " Fertialisation "

3.4 Expérimentation

Dans cette section nous décrivons les expériences menées pour analyser les performances de notre approche.

Expérimentation 1 :

Dans cette 1ère expérimentation (figure 3.8) nous voulons enregistrer les meilleurs scores de la fonction fitness en fonction des itérations. Elle est conduite avec les paramètres suivants :

– La taille de la population : 10

– Le nombre maximal de générations : 20.

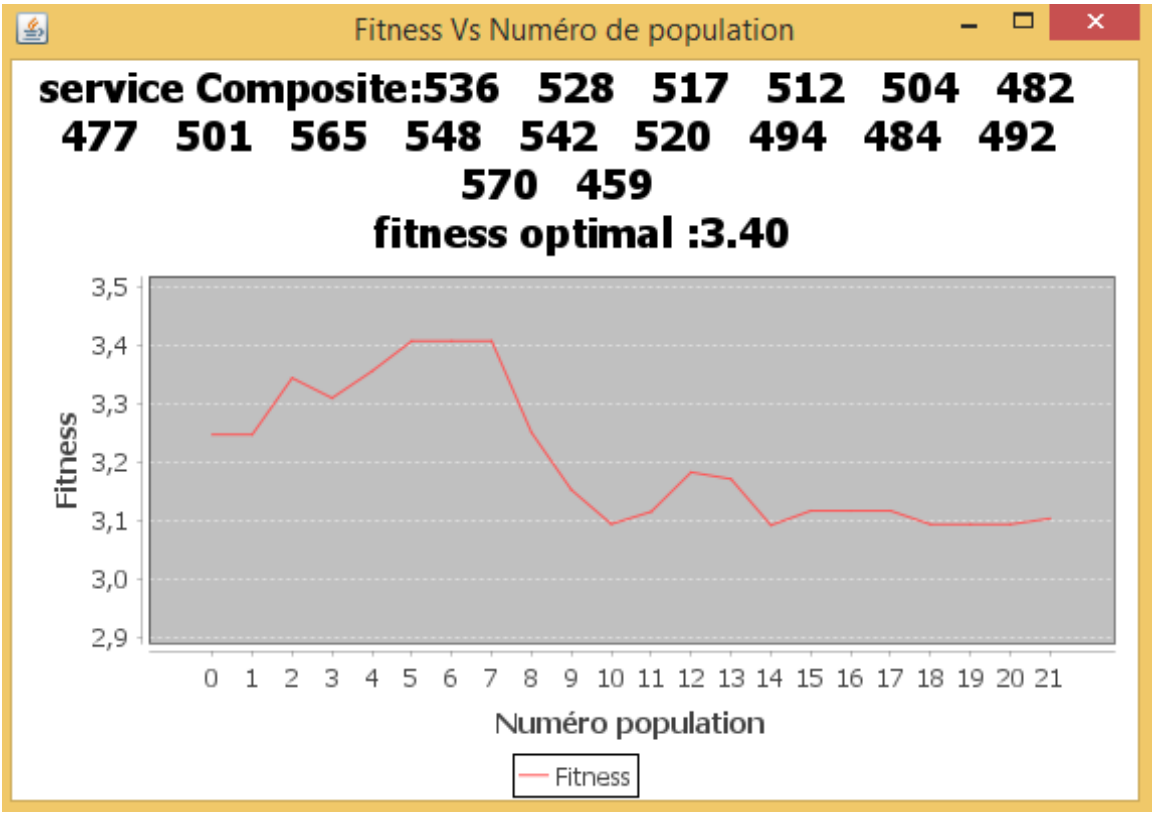

FIG. 3.9 – Changement Fitness dans tous les générations de AG

Expérimentation 2 :

Dans cette deuxième expérimentation (figure 3.9) nous voulons étudier l'impact de l'energie des services concrets avant et aprés le processus de composition.

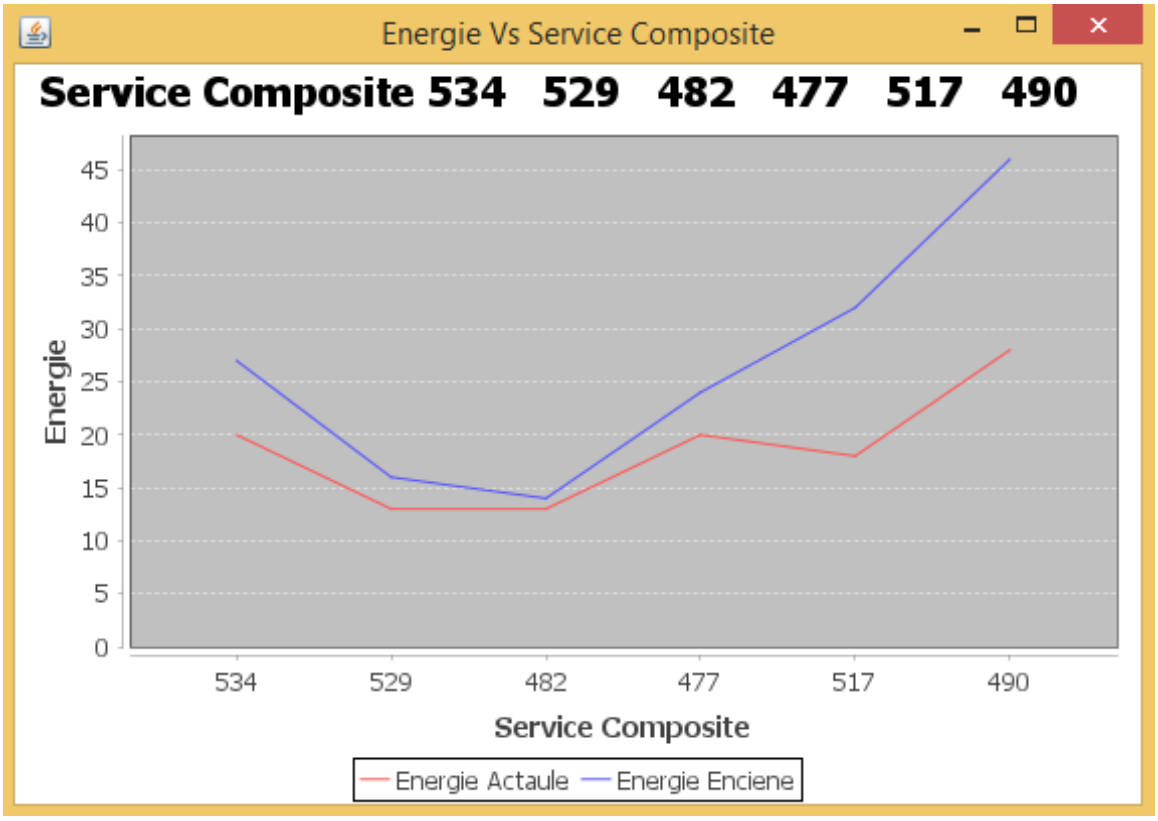

FIG. 3.10 – Changement L'energie avant et aprés le processus de composition

3.5 Conclusion

Ce chapitre est axé sur la validation et l'analyse de notre approche proposée. Les résultats expérimentaux montrent que l'algorithme proposé améliore le temps de sélection, cela est du principalement à la phase de sélection locale des services concrets et l'utilisation des algorithmes génétiques sur la phase de sélection globale. En outre, la solution proposée est proche de l'optimum en termes de QoS dans un temps d'exécution acceptable. Ensuite, nous avons présenté les différentes interfaces correspondantes à l'application.

Conclusion générale et perspectives

L'IoT devient de plus en plus répandu et adapté aux interactions des utilisateurs, qui composent les technologies de service de manière hétérogène. En règle générale, les appareils IoT sont intégrés de manière hétérogène et dynamique avec les fonctions de fiabilité liées à leur QoS pour chaque service atomique. Chaque service atomique est lié au dispositif IoT qui fournit la fonctionnalité du service. Dans la plupart des cas, un seul service n'est pas suffisant pour répondre aux exigences complexes de l'utilisateur.

Dans le cadre du présent mémoire, nous avons proposé une approche de composition de service IOT qui permet de choisir le meilleur service parmi les services disponibles selon la qualité de service pour satisfaire la requête de l'utilisateur en utilisant les algorithmes génétiques.

Notre application sélectionne les services composites les plus satisfaisantes, en se basant sur six(06) critères de QoS : Temps de réponse, fiabilité, disponibilité,energie,prix,localisation.

Au cours de ce mémoire, nous avons présenté au début un état de l'art sur les concepts relatifs au notre sujet : l'internet des objets, L'architecture orienté service (SOA) et composition des services IoT. Afin de modéliser notre proposition, nous avons commencé par la conception globale et conception détaillé de notre système en utilisant le formalisme UML et plusieurs fugires pour montrer notre approuche.

Afn de valider l'approche proposée, nous avons développé une application écrite en langage JAVA,les résultats expérimentaux obtenus sont illustrés dans le dernier chapitre du mémoire.

Perspectives

Dans le futur et comme perspectives de ce travail, nous considérerons les extensions suivantes :

- Combiner l'algorithme génétique avec d'autres techniques d'optimisation, par exemple les algorithmes de colonies de fourmis.
- Prendre en considération d'autres attributs à savoir l'environnement IOT (business ou économique , médicale..etc).
- Optimiser le plan de composition avec d'autre Caractristique à part la réputation.

Bibliographie

[1] Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012, December). Future internet : the internet of things architecture, possible applications and key challenges. In 2012 10th international conference on frontiers of information technology (pp. 257-260). IEEE.

[2] Jean-Michel, R. É. V. E. I. L. L. A. C. (2013). La réalité augmentée : techniques et entités virtuelles. Lavoisier.

[3] https ://www.postscapes.com/smart-greenhouses/ , page consultée le 08/05/2022

[4] Yılmaz, Ö., & Erdur, R. C. (2012). iConAwa–An intelligent context-aware system. Expert Systems with Applications, 39(3), 2907-2918.

[5] Han, M., & Zhang, H. (2013). Business intelligence architecture based on internet of things. Journal of Theoretical & Applied Information Technology, 50(1), 90-95.

[6] Lemoine, F. (2019). Internet des Objets centré service autocontrôlé (Doctoral dissertation, Conservatoire national des arts et metiers-CNAM).

[7] Meftah, Z. O. U. A. I. (2021). Une approche cloud computing basée IoT pour le smart House (Doctoral dissertation, Université de mohamed kheider biskra).

[8] Sheng, Z., Yang, S., Yu, Y., Vasilakos, A. V., McCann, J. A., & Leung, K. K. (2013). A survey on the ietf protocol suite for the internet of things : Standards, challenges, and opportunities. IEEE wireless communications, 20(6), 91-98.

[9] Mahmoud, R., Yousuf, T., Aloul, F., & Zualkernan, I. (2015, December). Internet of things (IoT) security : Current status, challenges and prospective measures. In 2015 10th international conference for internet technology and secured transactions (ICITST) (pp. 336-341). IEEE.

[10] Benghozi, P. J., Bureau, S., & Massit-Folléa, F. (2015). L'Internet des objets/The Internet of Things : Quels enjeux pour l'Europe ?/What Challenges for Europe ?. Les Editions de la MSH.

[11] https ://fr.wikipedia.org/wiki/Internet_des_objets, page consultée le 04/05/2022.

[12] Christian, Fnac, blog, http ://www.fnac.com/Avec-les-objets-connectes-le-changementcest-maintenant-MAJ-Mars-2017/cp20440/w-4, page consultée le 04/05/2022.

[13] Vermesan, O., & Friess, P. (Eds.). (2014). Internet of things-from research and innovation to market deployment (Vol. 29). Aalborg : River publishers.

[14] Vangelista, L., Zanella, A., & Zorzi, M. (2015, September). Long-range IoT technologies : The dawn of LoRa™. In Future access enablers of ubiquitous and intelligent infrastructures (pp. 51-58). Springer, Cham.

[15] Shah, S. H., & Yaqoob, I. (2016, August). A survey : Internet of Things (IOT) technologies, applications and challenges. In 2016 IEEE Smart Energy Grid Engineering (SEGE) (pp. 381-385). IEEE.

[16] Aoudia, I., Benharzallah, S., Kahloul, L., & Kazar, O. (2019). Service composition approaches for internet of things : a review. International Journal of Communication Networks and Distributed Systems, 23(2), 194-230.

[17] http ://www.service-architecture.com ,page consultée le 04/05/2022.

[18] Bossi, L., Braghin, S., & Trombetta, A. (2014, June). Multidimensional reputation network for service composition in the internet of things. In 2014 IEEE International Conference on Services Computing (pp. 685-692). IEEE.

[19] Manes, A. T. (2006). Service-Oriented Architecture : Developing the Enterprise Roadmap. Version, 2, 22.

[20] Yang, R., Li, B., & Cheng, C. (2014, November). Adaptable service composition for intelligent logistics : A middleware approach. In 2014 International Conference on Cloud Computing and Big Data (pp. 75-82). IEEE.

[21] Kellert, P., & Toumani, F. (2003). Les web services sémantiques. CLR03], chapitre, 7, 93-106.

[22] Benatallah, B., Dijkman, R. M., Dumas, M., & Maamar, Z. (2005). Service-composition : concepts, techniques, tools and trends. In Service-Oriented Software System Engineering : Challenges and Practices (pp. 48-67). IGI Global.

[23] Du, Y., Li, X., & Xiong, P. (2012). A Petri net approach to mediation-aided composition of web services. IEEE Transactions on Automation Science and Engineering, 9(2), 429-435..

[24] Li, L., Jin, Z., Li, G., Zheng, L., & Wei, Q. (2012, June). Modeling and analyzing the reliability and cost of service composition in the IoT : A probabilistic approach. In 2012 IEEE 19th International Conference on Web Services (pp. 584-591). IEEE.

[25] Rao, J., & Su, X. (2004, July). A survey of automated web service composition methods. In International Workshop on Semantic Web Services and Web Process Composition (pp. 43-54). Springer, Berlin, Heidelberg.

[26] Asghari, P., Rahmani, A. M., & Javadi, H. H. S. (2018). Service composition approaches in IoT : A systematic review. Journal of Network and Computer Applications, 120, 61-77.

[27] Yachir, A. (2014). Composition dynamique de services sensibles au contexte dans les systèmes intelligents ambiants (Doctoral dissertation, Université Paris-Est ; Université des sciences et de la technologie Houari Boumediene (Alger)).

[28] Kansal, A., & Zhao, F. (2008). Fine-grained energy profiling for power-aware application design. ACM SIGMETRICS Performance Evaluation Review, 36(2), 26-31.

[29] Lopez-Velasco, C. (2008). Sélection et composition de services Web pour la génération d'applications adaptées au contexte d'utilisation (Doctoral dissertation, Université Joseph-Fourier-Grenoble I).

[30] KOUICEM, A. (2008). Composition dynamique de services en environnement ubiquitaires (Doctoral dissertation, Université de Béjaia-Abderrahmane Mira).

[31] Saied, Y. B., Olivereau, A., Zeghlache, D., & Laurent, M. (2013). Trust management system design for the Internet of Things : A context-aware and multi-service approach. Computers & Security, 39, 351-365.

[32] Vallée, T., & Yildizoğlu, M. (2004). Présentation des algorithmes génétiques et de leurs applications en économie. Revue d'économie politique, 711-745.

[33] F. Laforest and F. Le Mou¨el. Syst'emes d'information pervasifs, Cours de master,Grenoble, 2006.

[34] https ://www.addsecure.fr/blog/utilisation-de-la-technologie-ido-dans-la-vie-de-tous-les-jours/,page consultée le 04/05/2022.

[35] Yang, R., Li, B., & Cheng, C. (2014, December). A petri net-based approach to service composition and monitoring in the IOT. In 2014 Asia-Pacific Services Computing Conference (pp. 16-22). IEEE.

[36] Zhang, L., Yu, S., Ding, X., & Wang, X. (2014, August). Research on IOT RESTful web service asynchronous composition based on BPEL. In 2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics (Vol. 1, pp. 62-65). IEEE.

[37] Guinard, D., Trifa, V., Karnouskos, S., Spiess, P., & Savio, D. (2010). Interacting with the soa-based internet of things : Discovery, query, selection, and on-demand provisioning of web services. IEEE transactions on Services Computing, 3(3), 223-235.