

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE ECHAHID HAMMA LAKHDAR –EL-OUED

FACULTE DES SCIENCES EXACTES

Mémoire de Fin d'Étude
MASTER ACADEMIQUE

Domaine: Mathématiques et Informatique

Filière: Informatique

Spécialité: Systèmes Distribués et Intelligence Artificielle

THEME

**Une approche basée agent mobile pour
l'internet des choses
(Smart business environnement)**

Présenté par:

Yakoub KHALDI

Devant le jury composé de :

Président :	Mr. OTHMANI Samir	Maitre Assistant B (Univ. El-Oued)
Promoteur:	Mr. NAOUI M ^{ed} Anouar	Maitre Assistant A "Maitre de recherche" (Univ. El-Oued)
Examineur:	Mr. MEFTAH Charf Eddine	Maitre de conférences B (Univ. El-Oued)

Année Universitaire : 2015-2016

Remerciements

Dieu dit:

(وَإِذْ تَأْتِيَنَّكَ رُبُكُمُ لِلَّذِينَ كَفَرْتُمْ إِنَّ عَذَابِي لَشَدِيدٌ)

(Verset 07 Sourate Ibrahim)

Louange à Dieu qui m'a montré le chemin de la science et de la connaissance et a m'a aidé à accomplir cette tâche et m'a aidé à finir ce travail, comme l'indique l'impact : qui ne pas remercier personnes ne pas remercier Dieu .

Que ce soit d'un point de vue scientifique ou humain, la réalisation de ce mémoire fut pour moi une expérience d'une immense valeur.

Je remercie vivement **Mr Med Anouar Naoui** , d'avoir accepté de me diriger tout au long de ce travail, qu'elle trouve ici le témoignage de ma gratitude.

Je remercie également chef de partement: **professeur MDILEH** Sur ses efforts
pour amener ce travail

Je remercie du fond du cœur mes formidables collègues et amis qui ont toujours été là pour me soutenir et avec qui j'ai partagé des moments inoubliables.

Dédicace

A mes très chers parents.

Pour ma mère chère

A mes très chères sœurs.

A mon très chère Frère.

Aucun mot ne pourra exprimer mes sentiments envers vous.

A toute ma famille.

A tous mes chers amis : avec tous mes souhaits de réussir,

Pour tout votre soutien et votre amitié je vous dis MERCI.

A tous ceux qui m'aiment

A tous ceux que j'aime

Je dédie ce travail...

Sommaire

Remerciements	
Dédicace	
Sommaire	
Résumé	
INTRODUCTION GENERAL	1
CHAPITRE 01 : L'approche agent mobile	
Introduction	3
1.1 Pourquoi les agents mobiles	4
1.2 Structure d'un agent mobile	4
1.3 La mobilité des agents	5
1.4 Domaines d'application	5
1.5 La sécurité de l'agent mobile	6
1.6 Présentation de la technologie d'agent mobiles	7
1.7 Comparaison entre les paradigmes "Client/ Serveur" et "Agent mobile"	9
1.8 Composition d'un modèle d'agents mobiles	10
1.9 Agents mobiles	10
1.10 Le Serveur d'agent/ système d'agent	12
1.11 Migration d'un agent	12
1.12 Tolérance aux pannes	13
1.13 Plate-forme et Standardisation	13
1.14 Les normes	14
1.15 Implémentations existantes	15
1.16 Avantages de la technologie par Agents Mobiles	15
Conclusion	16
CHAPITRE 02 : Internet des Objets	
INTRODUCTION	18
1. Définition de l'internet des objets	19
2. Historique de l'internet des objets	19
3. Les capteurs composants majeurs de l'internet des objets	20
4. Les services de l'Internet et l'IoT	22

5. Les protocoles à l'œuvre dans l'Internet des objets	26
6. Quelles solutions architecturales pour l'IoT	28
Conclusion	31

CHAPITRE 03 : Modélisation

1. Introduction	33
2. Les Travaux similaires	33
3. environnement Distribué dans IoT	35
4. Diagramme d'Activité	40
5. Diagramme de Séquence	41
6. Diagramme de class	42
Conclusion	42

CHAPITRE 04 : Implémentation

INTRODUCTION	44
1. QU'EST-CE QU'ARDUINO	44
2. POURQUOI ARDUINO	44
3. Travaux pratiques	45
Conclusion	50
CONCLUSION GENERAL	52
Bibliographie	

Liste des Figures

Chapitre 01

Fig1.1 - Le paradigme des agents mobiles.....	07
Fig 1.2 - Client/ Serveur Versus Agent mobile.....	09
Fig 1.3 - Un simple modèle d'agents mobiles.....	10
Fig1.4 - Le paradigme agents mobiles.....	11

Chapitre 02

Fig 2.1 - Panorama des capteurs utilisés dans le laboratoire LIGM.....	20
Fig 2.2: Capteurs sur un billet de 1 dollar utilisé comme échelle ce qui montre qu'ils sont très petits.....	21
Fig 2.3 - L'Architecture Orientée Service (SOA).....	23
Fig 2.4 - Quelques objets intelligents mis en œuvre dans l'Internet des objets.....	25

Chapitre 03

Fig 3.1 IoT architecture basée sur des IoT SOA.....	36
Fig 3.2 Architecture IoT SOA propos.....	37
Fig 3.3 couches Environnement.....	38
Fig 3.4 Architecture globale	39
Fig 3.5 Diagramme d'Activité.....	40
Fig 3.6 Diagramme de Séquence.....	41
Fig 3.7 Diagramme de classes.....	42

Chapitre 04

Figure 4.1 interface de programme.....	46
Figure 4.2 Circuit de notre programme de détection des obstacles.....	46
Figure 4.3 Carte Arduino et câblage de détection des obstacles.....	47
Figure 4.4 Agent mobile (Panier).....	47

Figure 4.5 le programme de détection de mouvement.....	48
Figure 4.6 Carte Arduino et câblage de détection des mouvement.....	49
Figure 4.7 Agent mobile détection des mouvement.....	49

الملخص:

قمنا في هذه المذكرة بدراسة مفهوم أنترنت الأشياء المعتمدة أساسا على العون المتحرك وركزنا أيضا على مفهوم البيئة حيث قمنا بتصميم نموذج يأخذ بعين الاعتبار المحيط وبذلك نتمكن من تسهيل عملية إنشاء نظم موزعة خاصة بأنترنت الأشياء.

الكلمات المفتاحية: أنترنت الأشياء - العون - المتحرك.

Résumé:

Dans cette note, nous étudions le concept de l'Internet des choses, basé essentiellement sur le paradigme agent mobile. Aussi, nous nous sommes concentrés sur le concept de l'environnement où nous avons conçu un modèle qui prend en compte ce concept, afin de faciliter le processus d'implémentation des systèmes IoT dans un environnement distribué.

Mots clés : Internet des choses - agent – mobile.

Abstract :

We are study the Internet of things based essentially on mobile agent paradigm. Also we focused on the concept of the environment we have designed a model that takes into account this concept, and so we can facilitate the process of creating distributed systems to the Internet of things.

Keywords: Internet of things - agent - mobile.

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Internet des objets (IoT) est une nouvelle technologie qui offre au système informatique, un système non centre utilisateur seulement, mais prendre en compte un objet et leur gestion.

Cette nouvelle discipline ouvre plusieurs axe de recherche tel que l'architecture d'IoT , la modélisation des objets.

Notre travail propose une architecture IoT et un modèle a base d'agent.

- architecture IoT : cette architecture comble les insuffisantes des architecture de système par la modélisation d'environnement et ses objets.
- Notre modèle a base d'agent pour modéliser les aspect d'autonomie et intelligence collective.
- Notre mémoire est organise en 4 chapitres :

La première chapitre présente La technologie agent , la deuxième chapitre présent internet des objet et 3 et 4 chapitre la modélisation et l'implémentation.

CHAPITRE 01

L'approche agent mobile



Chapitre 01:L'approche agent mobile

Introduction

Avant d'aborder le sujet de **l'Internet des objets** « Internet of Things » qui est basé sur le concept agent mobile, il faudrait d'abord parler sur ce concept d'agent mobile.

Ce concept est survenu pour remédier les insuffisances constatées dans le fonctionnement du paradigme client/serveur.

Je vais tout d'abord faire un petit tour sur les premiers remèdes affectés au paradigme client/serveur tels que « les appels de procédures à distance :RPC (Remote Procedure Call) », ou la sous programmation » .

Puis je vais entamer directement les agents mobiles : leur impérativité, leur structure, leur mobilité, leurs domaines d'applications, leur sécurité, et enfin leur technologie.

1.1 Pourquoi les agents mobiles

Les agents mobiles sont une catégorie particulière des agents logiciels dont la caractéristique prédominante est leur capacité de se transporter entre les nœuds d'un réseau ou de plusieurs réseaux [1]. Les agents mobiles sont une extension directe du paradigme client/serveur.

Dans ce paradigme, un serveur offre des services demandés par le client alors que le client procède à leur utilisation. La communication qui prend place entre le client et le serveur est basée sur un échange de messages, obligeant les programmeurs à gérer divers aspects liés à la communication comme les adresses réseaux et les points de synchronisation.

Ainsi le système des appels de procédures à distance (RPC : Remote Procedure Call) essaie de réduire les détails de la communication, en permettant au client d'exécuter un service distant de la même façon qu'un appel de fonctions locales. L'emplacement du serveur, l'initiation du service et le transport des résultats sont pris en charge au profit du client de manière transparente.

Mais malgré l'introduction du système RPC, un problème fondamental persiste dans les architectures client/serveur. Si le serveur ne fournit pas le service que le client exige, alors ce dernier doit réaliser plusieurs appels à distance afin d'obtenir le service requis.

Ceci peut résulter en une augmentation des temps d'attente et une transmission d'informations intermédiaires non utiles et non efficaces, ou voire même un blocage.

Une autre tentative de résolution de ce problème de fonctionnement des architectures client/serveur est l'introduction de la sous-programmation [1]. Elle permet aux clients d'exécuter leurs programmes dans le nœud où le service réside. De cette manière, les requêtes peuvent être initiées localement et les programmes peuvent traiter les résultats intermédiaires avant de transmettre les résultats finaux au client une fois le tout terminé.

Bien que les programmes puissent migrer de leurs sites originaux, ils ne peuvent pas se déplacer une deuxième fois vers d'autres systèmes ou ressources.

De plus, ces programmes sont généralement écrits pour des clients spécifiques, rendant ainsi leur réutilisation difficile.



Chapitre 01:L'approche agent mobile

La technologie des agents mobiles tente d'adresser les diverses insuffisances liées aux paradigmes client/serveur et de la sous-programmation. Les caractéristiques typiques des agents mobiles sont leurs capacités de migrer quand ils veulent, leur autonomie dans leurs actions, leurs liens avec les autres agents et leur indépendance par rapport à leur emplacement original [1].

1.2 Structure d'un agent mobile

Un agent mobile est une entité qui possède cinq attributs : son état, son implémentation, son interface, son identifiant et son autorité. Quand un agent se déplace à travers le réseau, il transporte ses attributs [2] :

- **L'état**: l'état d'un agent peut être considéré comme une photo instantanée de son exécution. Quand un agent voyage, il transporte avec lui son état, ceci lui permet de reprendre son exécution quand il a arrivé à destination.
- **L'implémentation**: comme n'importe quel autre programme, l'agent mobile a besoin d'un code pour pouvoir s'exécuter. Quand il se déplace à travers le réseau, l'agent peut soit emporter son code soit aller à destination, voir quel code est disponible sur la machine distante et récupérer le code manquant à partir du réseau (c'est la technique du « code on demande »).
- **L'interface**: un agent fournit une interface qui permet aux autres agents et autres systèmes d'interagir avec lui. Cette interface peut être un ensemble de méthode qui permet aux autres agents et applications d'accéder aux méthodes de l'agent par un système de messagerie.
- **L'identifiant**: chaque agent possède un identifiant unique durant son cycle de vie, qui lui permet d'être identifié et localisé. Puisque l'identifiant est unique, il peut être utilisé comme clé dans les opérations qui exigent un moyen pour référencer une instance particulière d'agents.
- **L'autorité** : une autorité est une entité dont l'identité peut être authentifiée par n'importe quel système auquel elle essaye d'accéder. Une autorité peut être soit une personne privée, soit une organisation. L'identité est constituée d'un nom et d'autres attributs.

1.3 La mobilité des agents

La mobilité est une caractéristique désirable dans les agents pour plusieurs raisons [1] :

- **Efficacité** : si un agent peut se diriger vers des emplacements désirés, alors le trafic réseau pourra être réduit. L'agent peut traiter l'information et décider si elle est importante à transférer. L'aspect de transfert est crucial quand il s'agit de situations où la connexion réseau est de faible débit.
- **Persistence** : une fois qu'un agent mobile est mis en place, il n'est pas dépendant du système qui l'a initié et ne doit pas être affecté si celui-ci venait à tomber en panne. La capacité de se déplacer entre les nœuds des réseaux donne à l'agent la capacité de "survivre" et d'atteindre le plus de ressources possibles. Ceci est particulièrement utile pour les usagers ayant des ordinateurs mobiles; ces usagers peuvent se connecter, initier l'agent, se déconnecter et contrôler de façon intermittente son progrès.
- **Communication "peer-to-peer"** : un inconvénient majeur du paradigme client/serveur est la non capacité des serveurs à communiquer entre eux. Les agents mobiles sont considérés comme des entités paritaires ("peer") et peuvent ainsi adopter la position la plus appropriée à leurs besoins actuels, en étant des clients ou des serveurs.



Chapitre 01:L'approche agent mobile

- Tolérance aux fautes: dans une situation d'échec du serveur ou du réseau pendant une requête, il est difficile pour le client de retrouver la situation initiale afin de se synchroniser de nouveau avec le serveur. Les agents mobiles n'ont pas besoin de maintenir des connexions permanentes, étant donné que leurs états d'exécution sont centralisés à l'intérieur d'eux.

1.4 Domaines d'applications

Les applications qui utilisent la mobilité comme un mécanisme pour s'adapter aux changements de ressources doivent satisfaire trois exigences. Premièrement, ces applications doivent être conscientes de leurs environnements d'exécution. En particulier, elles ont besoin de contrôler le niveau et la qualité des ressources que leurs environnements disposent. Deuxièmement, elles ont besoin d'être capables de réagir aux changements de la disponibilité des ressources. Troisièmement, elles doivent être capables de contrôler la manière avec laquelle les ressources sont utilisées en leurs noms.

Il existe plusieurs applications pour lesquelles les agents mobiles pourraient être utilisés. La majorité de ces applications s'intéresse à la recherche d'informations au nom de l'utilisateur et possiblement à l'exécution de transactions spécifiques quand les informations appropriées sont rencontrées. Voici ci-dessous une liste d'applications susceptibles d'utiliser le paradigme des agents mobiles comme mécanisme de fonctionnement [3,4].

- Collecte de données de plusieurs places : une différence majeure entre le code mobile, comme les applets, et les agents mobiles est l'itinéraire. Alors que le code mobile voyage d'ordinaire d'un point A à un point B, les agents mobiles ont un itinéraire plus complexe et peuvent voyager de manière séquentielle entre plusieurs sites. Une application naturelle par conséquent est la collecte d'informations localisées dans plusieurs ordinateurs d'un réseau.
- Recherche et filtrage : étant donné le volume important (et en croissance) d'informations disponibles sur l'Internet, la collecte d'informations exige la recherche dans un grand espace de données pour déterminer des portions d'informations pertinentes. Le filtrage des informations non-pertinentes est souvent un processus consommateur de temps. À la demande de l'utilisateur, un agent mobile pourrait visiter plusieurs sites, chercher à travers les informations disponibles dans chaque site et construire un index des liens pour les portions d'information qui répondent aux critères de sélection.
- Négociation : au lieu de chercher dans des bases de données ou dans des fichiers, les agents peuvent obtenir de l'information en interagissant avec d'autres agents. Si par exemple une personne désire organiser une rencontre avec diverses personnes, elle pourrait envoyer son agent mobile pour interagir avec les agents représentatifs de chacune de ces personnes. Les agents pourraient négocier et décider du temps et du lieu de la rencontre selon les contraintes de chaque personne. Ces contraintes sont exprimées dans un langage de spécification compréhensible aux agents.
- **Le commerce électronique** : C'est une application qui se déroule au sein d'un réseau de places de marché électroniques où des agents se rencontrent afin de proposer, rechercher et négocier différents types de services, pour le compte d'utilisateurs individuels ou de sociétés (réservations, achat de biens divers...). Ce type d'application, met en évidence deux types d'agents : agents vendeurs et agents acheteurs. Un agent vendeur est un agent chargé de vendre un produit. Il voyage de serveur en serveur à la recherche d'agents acheteurs de ce même produit. Une fois une rencontre effectuée, un algorithme de marchandage peut se mettre en œuvre. Si un accord d'achat est trouvé, il est réalisé directement ou/et un courrier électronique est envoyé aux propriétaires de ces agents.
- **La recherche d'informations sur le Web** : La quantité d'informations disponible sur le Web croît sans cesse que les outils de recherches disponibles deviennent inadaptés. Ainsi la recherche de l'information pertinente sur le Web devient une tâche pénible et ennuyeuse. Une



Chapitre 01:L'approche agent mobile

solution consiste en la délégation de cette mission à un ou plusieurs agents qui visitent les sites Web, et qui coopèrent et recherchent les sites d'intérêts, puis rentrent avec les meilleurs résultats.

- **L'équilibrage de charge dynamique** : En parallélisme, dans les machines de type MIMD, l'enjeu principal dans l'écriture d'un programme efficace est l'équilibrage de charge. Cet équilibrage est difficile à résoudre sur des problèmes très irréguliers. Les agents mobiles permettent d'équilibrer la charge d'une machine : si un nœud est en avance sur son travail tandis qu'un autre est en retard, un ou plusieurs agents peuvent migrer du dernier au premier réduisant ainsi l'écart.
- **Les mises à jour** : Le problème de la mise à jour des multiples copies, dans certains cas, peut être résolu par un agent dont le rôle est d'apporter la modification de site en site, selon un chemin prédéterminé.

1.5 la sécurité de l'agent mobile

La sécurité dans les systèmes d'agents mobiles touche deux parties [5]. La sécurité du système contre les atteintes d'un agent dangereux et malveillant, et la sécurité d'un agent contre un système qui tente de l'espionner, de lui voler des informations confidentielles ou de changer son code d'exécution. Dans ce cas, l'agent pourrait fonctionner au nom du système alors que les frais de traitement seraient chargés au propriétaire original de l'agent.

Dans les systèmes d'agents mobiles, les questions liées à la sécurité sont très diverses. Elles incluent : (1) la communication entre les plates-formes doit être sécurisée, (2) les agents entrants doivent être authentifiés et autorisés à exécuter des opérations en fonction de leurs identités, (3) toutes les interactions entre l'infrastructure de support des agents et les ordinateurs hôtes doivent être contrôlées et vérifiées, (4) l'infrastructure doit être protégée des agents malveillants et (5) les agents doivent être protégés entre eux.

Le Tableau suivant reprend les divers risques ainsi que les techniques de défense généralement utilisées.

Composantes de la sécurité dans les systèmes d'agents mobiles

Réseau	accès à des informations privées encapsulées dans un agent touchant à son état	techniques de cryptographie, authentification, canaux sécurisés
Système hôte	accès non-autorisé à des ressources locales	& accès au hôte accordé par l'infrastructure selon une base discriminatoire
Plate-forme	Interférence dans l'état de l'infrastructure en lisant, écrivant ou exécutant du code	communication agent-infrastructure sécurisé et comptabilisation de l'usage des ressources
Agent	Interférence dans l'état de l'agent en lisant, écrivant ou exécutant du code	communication inter-agents sécurisée et comptabilisation de l'usage des ressources

La sécurité du réseau de communication et de l'authentification est un problème bien étudié et souvent résolu par des techniques de cryptographie. La protection du système hôte et des droits d'accès a une grande similarité avec la protection des systèmes d'exploitation. La difficulté est de définir les ressources à protéger afin de mettre en place des mécanismes de contrôle d'accès flexibles. La protection des infrastructures de support des agents peut être



Chapitre 01:L'approche agent mobile

réalisée en imposant certaines restrictions aux programmes, comme l'interdiction de manipuler arbitrairement les zones mémoires. La communication entre les agents est aussi importante, étant donné que les agents ont le pouvoir de communiquer librement, tout en leur interdisant d'interférer dans les opérations d'exécution des autres agents.

Un agent peut être soumis à quatre catégories d'attaques [6] : violation de la partie privée, violation de l'intégrité, "masquera ding" (Usurpation de l'autorité ou de l'identité d'un autre agent pour une série d'actions) et dénégation d'un service (Consommation excessive d'une ressource finie comme du temps de traitement, de la mémoire ou du sous-système de communication).

1.6 Présentation de la technologie d'agents mobiles [8] :

Le client donne une mission à un agent. Pour la réaliser, l'agent se déplace dans le réseau de Machines accédant localement aux services offerts par ces machines. On peut distinguer trois phases :

- L'activation de l'agent mobile avec la description de sa mission.
- L'exécution de la mission par l'agent qui se déplace pour accéder aux services.
- La récupération éventuelle des résultats de l'agent mobile.

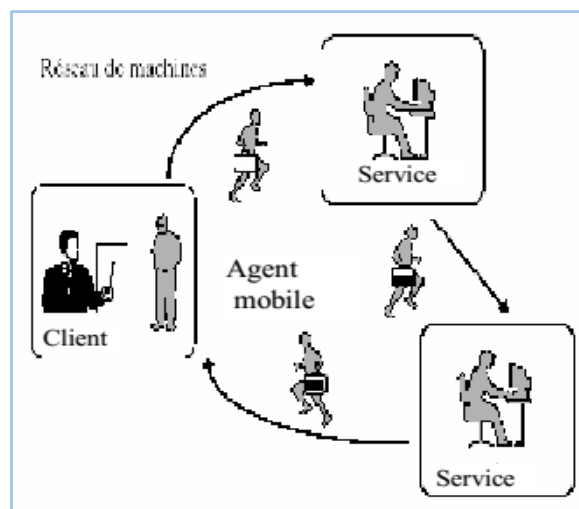


FIGURE 1.1 - Le paradigme des agents mobiles

1.6.1 Caractéristiques

Les différentes caractéristiques de l'entité agent peuvent être résumées dans la définition suivante :

« un agent est un système informatique situé dans un environnement, et qui agit de façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu » [7].

1.6.1.1 L'autonomie

C'est la caractéristique la plus intéressante de l'agent. En effet, l'agent est capable d'agir sans aucune intervention et il contrôle son comportement ainsi que son état interne, autrement dit, l'agent est dit autonome dans le sens où le concepteur du système ne pilote pas son



Chapitre 01:L'approche agent mobile

comportement c'est-à-dire l'agent décide lui-même quelle action à entreprendre parmi celles qui sont lui possibles.

L'autonomie de l'agent rend son comportement indéterministe, c'est-à-dire aucun autre agent ne peut observer ou modifier sa partie décisionnelle et donc prédire ses comportements. L'agent choisit son comportement en fonction de ces perceptions, il se comportera différemment en fonction de ces dernières. L'agent peut être vu comme une boîte noire, on ne peut observer que les comportements externes.

1.6.1.2 La situation

Les agents sont situés dans un environnement contenant également des entités passives, manipulées par les agents (par exemple : des ressources, des données, des objets physiques...) et communément appelées objets.

L'agent est capable d'agir sur son environnement qu'il peut percevoir grâce à ses entrées sensorielles.

L'agent doit s'adapter sans cesse aux changements de son entourage qui pourraient modifier de façon pertinente son comportement à tous les niveaux (objectif, plan, action...etc.).

1.6.1.3 La flexibilité

L'agent est capable de réaliser des actions d'une façon autonome et réflexible afin d'atteindre les objectifs qui lui ont été fixé.

La réflexibilité signifie dans ce cas :

- la réactivité: l'agent est capable de percevoir les changements dans son environnement, et doit élaborer une réponse dans les temps requis (changer son état interne, exécuter une tâche particulière...etc.).
- la pro-activité: l'agent doit exhiber un comportement proactif, opportuniste et orienté objectifs, c'est-à-dire il n'agit pas uniquement en réponse à son environnement mais, il est également capable de prendre l'initiative au "bon" moment.
- la sociabilité: l'agent doit être capable d'interagir avec les autres agents (logiciels et humains) quand la situation l'exige afin de compléter ses tâches ou aider ces agents à accomplir les leurs.

1.6.2 Motivations d'utilisations du paradigme d'agents mobiles

Plusieurs raisons peuvent expliquer l'engouement soudain pour cette technologie.

- × De nombreux problèmes sont par nature distribués.
- × les technologies réseau ont reçu de considérables attentions de la part des chercheurs et des techniciens, rendant ainsi la généralisation des agents mobiles envisageable.
- × La vitesse des transmissions, la sécurité des communications et la standardisation des protocoles rendent plus facile leur mise en œuvre.
- × L'accroissement du pouvoir d'expression des calculs formels autorise la modélisation d'architectures distribuées indispensable à la notion d'agents mobiles.



Chapitre 01:L'approche agent mobile

- × D'autre part, des langages de haut niveau (Java, C++, ...) permettent l'écriture rapide de système d'agents mobiles.
- × En plus, la migration des agents mobiles à des sites distants permet la confrontation de plusieurs experts, permettant ainsi de bénéficier et d'apprendre de leurs savoir-faire.

1.7. Comparaison entre les paradigmes "Client/Serveur" et "Agent mobile"[9]

Dans le paradigme : "client/serveur", un client demande un service auprès du serveur, d'une façon interactive et synchrone, ceci signifie que le client est bloqué tant que le serveur n'a pas répondu. Tandis qu'au niveau du modèle "agent mobile", l'agent se déplace d'une façon autonome entre les machines d'un réseau pour réaliser certaines tâches localement. Ces agents s'exécutent d'une façon asynchrone ce qui permet de dire que ce paradigme est mieux adapté que le client/serveur à des traitements longs et nécessitant des interrogations fréquentes du serveur.

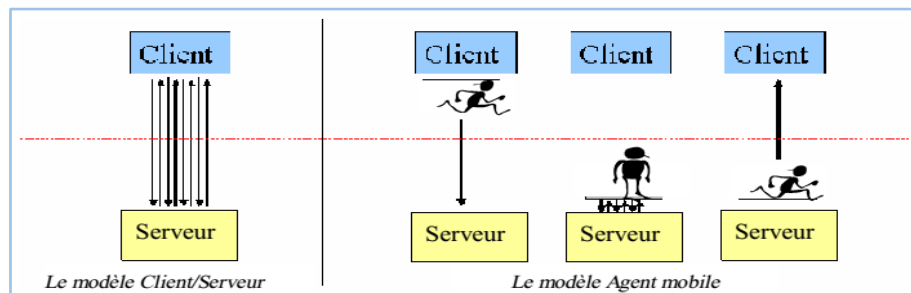


FIGURE1.2- Client/Serveur Versus Agent mobile.

1.7.1. Intérêts des agents mobiles

Parmi les intérêts de l'utilisation des agents mobiles:

- Réduction de la charge dans le réseau : Dans le cas des applications faisant des interrogations très fréquentes de la base de données, il serait certainement moins coûteux de faire migrer le code de traitement que les données à traiter, qui peuvent être beaucoup plus volumineuses. Ceci induit une minimisation des communications distantes.
- Exécution asynchrone et autonome : Le client peut faire autre chose pendant que l'agent mobile accomplit sa mission -voire même être éteint-.
- Efficacité & souplesse du traitement : La mobilité permet plus d'efficacité et de souplesse, en privilégiant les interactions locales évitant ainsi le transfert de données intermédiaires et permettant plus de confidentialité de traitement.
- Adaptation dynamique à l'environnement : Les agents mobiles adaptatifs savent détecter les changements dans leur environnement, y réagir de manière autonome et s'adapter en conséquence en se déplaçant vers des endroits plus rationnels pour leurs tâches.
- Robustesse et tolérance aux fautes :Lorsqu'un système/machine hôte est en difficulté (rupture de communication,...), les agents mobiles visiteurs prévenus ont la possibilité de se dispatcher ailleurs dans le réseau.



Chapitre 01:L'approche agent mobile

- Performance : La nature du paradigme d'agent mobile supportant la décomposition du travail, la modularité, la réutilisabilité et le parallélisme, procure un traitement performant et compétitif. Le système est plus robuste car réparti et parallélisé. De plus, une partie de l'application s'exécute dans le réseau et peut bénéficier des ressources importantes.

1.8. Composition d'un modèle d'agents mobiles

Un modèle à agents mobiles comprend deux concepts fondamentaux ; l'agent et son environnement appelé système d'agents ou encore plateforme d'agents.

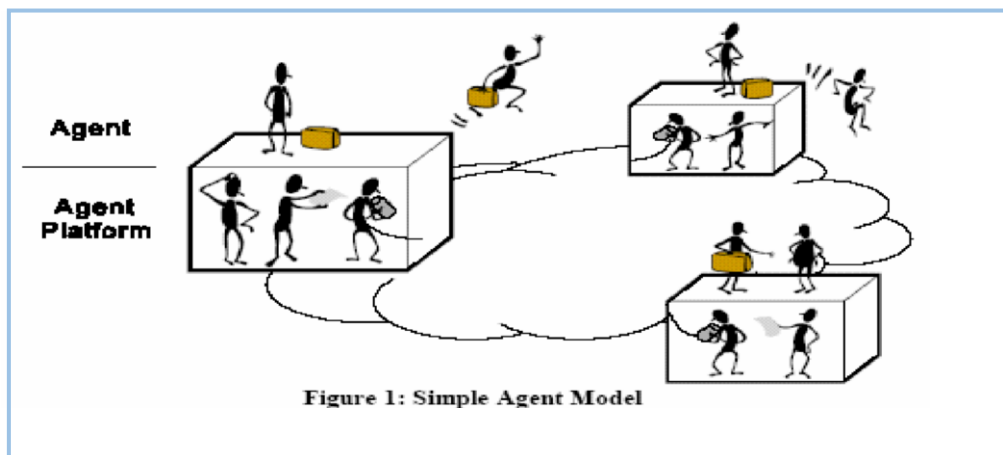


FIGURE 1.3 - Un simple modèle d'agents mobiles

1.9. Agents mobiles [10]

Les agents mobiles sont des entités logicielles qui peuvent se déplacer dans le réseau de leur propre initiative ; ils se déplacent d'une machine à une autre et communiquent avec d'autres agents ou accèdent aux ressources du serveur. Les agents mobiles ont suscité un grand intérêt pendant les dernières années pour leur capacité à supporter les interactions asynchrones et à réduire le trafic dans le réseau pendant les interactions client/serveur [6]. Dans la plupart de cas, les systèmes d'agents mobiles fournissent un ensemble de services de haut niveau tels que les fonctions de surveillance, la planification de voyage, la gestion de ressources, des mécanismes d'introspection et des langages de communication de haut niveau. La technologie d'agents mobiles permet aussi de:

- Fournir une exécution asynchrone de tâche. Ainsi, la dépendance entre les clients et les applications de serveur peut être réduite et un traitement automatique de tâche est présenté.
- Déplacer un programme vers un serveur à distance pour le développement d'un nouveau type d'applications réparties. Les services ne sont plus liés à un certain environnement. Au lieu de cela, ils peuvent être dynamiquement installés et employés dans l'endroit exact où ils sont exigés.
- Utiliser la technologie d'agent mobile pour aider à réaliser une meilleure exploitation de ressource. En transférant des applications du client au serveur et en exécutant des appels locaux de procédure au lieu des appels extérieurs, le trafic réseau est réduit.



Chapitre 01:L'approche agent mobile

Les principaux attributs d'un agent mobile sont la mobilité du code, des données et de l'état (état d'un processus, d'une machine ou d'un protocole). Ceci permet à des entités logicielles de se déplacer de façon autonome à travers le réseau pour accomplir des tâches spécifiques, tirant profit de ce fait de la proximité.

Mais les avantages principaux des agents mobiles sont dus à la mobilité de code qui permet de réaliser des interactions asynchrones et de réduire le coût de communication.

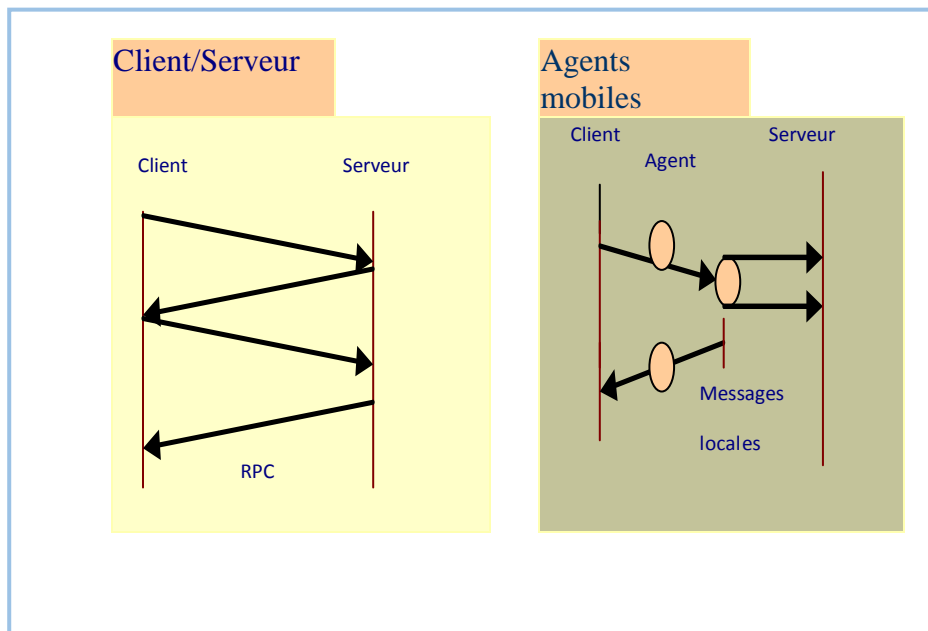


FIGURE 1. 4. Le paradigme agents mobiles

L'informatique répartie traditionnelle est généralement basée sur le modèle client/serveur qui est fréquemment lié au mécanisme de RPC (Remote Procedure Call).

Quand un RPC est appelé, des paramètres sont envoyés au serveur pour exécuter la procédure et le résultat est renvoyé au client. Comme décrit dans la partie gauche de la figure 2, chaque RPC exige l'établissement d'un canal de communication.

La partie droite de la figure 2 montre l'utilisation des agents mobiles ; la mobilité de code permet à une partie du programme client de se déplacer vers le serveur, d'agir localement et de revenir au client, une fois les interactions exécutées. Les améliorations d'exécution dépendent de la bande passante dans le réseau et du nombre de messages qui peuvent être optimisés au passage sur le réseau.

Il y a un grand nombre d'applications qui peut profiter des avantages de l'utilisation des agents mobiles. Les agents mobiles peuvent être une solution élégante pour le commerce électronique, pour la recherche dans les bases de données, la gestion de réseau et le calcul mobile.

Enfin, la sécurité est toujours le point faible des systèmes d'agents mobiles et c'est un domaine prometteur de recherche.



Chapitre 01:L'approche agent mobile

1.10. Le Serveur d'agent/système d'agent [9]

a. Définition d'un serveur d'agents : Un « serveur d'agents » ou encore une « agence » est le contexte d'exécution d'agents, là où un agent exerce son activité et le destinataire d'une éventuelle migration ne peut-être qu'un serveur d'agents.

b. définition d'un système d'agents : Un système d'agents est une plateforme qui peut créer, interpréter, exécuter, transférer et terminer un agent. Il peut être un réseau de serveurs d'agents. Autrement dit un système d'agents est la vision utilisateur de l'ensemble des serveurs et des agents sur un réseau.

Notons que les termes de serveur d'agents, système d'agents et réseau de système d'agents sont souvent confondus dans la littérature.

De la même manière qu'un agent, un système d'agent est associé à une autorité qui identifie la personne ou l'organisation pour laquelle il agit. Un système d'agent est identifié par son nom et son adresse. Une machine hôte peut contenir plusieurs systèmes, d'agent. Cinq concepts jouent un rôle important dans le système d'agent.

- La place : Une place est un contexte au sein d'un système d'agent, dans lequel un agent s'exécute. Ce contexte peut fournir un ensemble de services uniformes sur lesquels l'agent peut compter indépendamment de sa localisation spécifique. La place de départ et la place de destination peuvent être situées au sein d'un même système d'agent ou sur des systèmes d'agents différents. Un système d'agent peut contenir une ou plusieurs places.

- Le type du système d'agent: Le type d'un système d'agent permet de définir le profil d'un agent. Par exemple, si le type d'un système d'agents est «AGLET", alors le système d'agent est implémenté par IBM et supporte Java comme langage de programmation.

- Les ressources: Le système d'agent et l'a placé fournissent un accès contrôlé aux ressources locales et aux services (base de données, processeurs, mémoire, disques).

- La localisation: On définit la localisation d'un agent comme étant la combinaison de la place dans laquelle il s'exécute et l'adresse réseau du système d'agent où réside la place. Concrètement, elle est définie par l'adresse IP et le port d'écoute du système d'agent avec le nom de la place comme attribut.

- La région : Une région est un ensemble de systèmes d'agent qui appartiennent à la même autorité mais qui ne sont pas nécessairement de même type.

1.11. Migration d'un agent

Les migrations d'agents mobiles peuvent s'effectuer selon deux modes :

1.11.1 Migration forte

La migration forte, où la totalité de l'agent (c'est-à-dire code, données et unité d'exécution) migre vers le nouveau site. Pour cette migration réelle, l'agent est suspendu ou capturé avant d'être transféré. Une fois arrivé sur le site distant, il redémarre son exécution au point de contrôle précédent, en conservant l'état du processus. Une autre possibilité proposée consiste à



Chapitre 01:L'approche agent mobile

stopper l'exécution de l'agent avant la migration puis d'en créer une copie distante identique sur le site distant (migration par réplication).

1.11.2. Migration faible

La migration faible ne fait que transférer avec l'agent son code et ses données. Sur le site de destination, l'agent redémarre son exécution depuis le début en appelant la méthode qui représente le point d'entrée de l'exécution de l'agent, et le contexte d'exécution de l'agent est réinitialisé.

1.12. Tolérance aux pannes

Le modèle d'exécution de l'agent mobile implique son interaction avec plusieurs sites ce qui expose l'agent à une éventualité de disparition à cause de la défaillance ou de la déconnexion soudaine et imprévue d'un site sur lequel il s'exécute. La disparition d'un agent entraîne un dysfonctionnement de l'application basée sur ce dernier. Une application distribuée sûre doit pouvoir continuer de fonctionner en cas de défaillance d'une partie du système. Pour certains types d'applications, il est essentiel que les environnements d'exécution d'agents mobiles offrent des mécanismes de tolérance aux fautes.

Dans une architecture de services, les défaillances de sites peuvent conduire à un comportement défaillant d'un service et donc le rendre inutilisable par le client. Plusieurs types de défaillances sont à considérer [11]:

- Une défaillance par arrêt (crash, panne) quand un serveur ne rend pas de résultats suite à des invocations répétées.
- Une défaillance par omission quand un serveur omet de répondre à son client.
- Une défaillance temporelle quand la réponse du serveur est fonctionnellement correcte mais n'est pas arrivée dans un intervalle de temps donné.
- Une défaillance de valeur quand le serveur rend des résultats incorrects.

Pour une application distribuée basée sur le concept d'agents mobiles, la migration des agents engendre d'autres types de défaillance :

- Un agent peut disparaître après avoir visité plusieurs sites. Si aucune précaution n'est prise, les résultats de son exécution sur ces sites peuvent être perdus.
- Il est important de détecter la disparition d'un agent pour en informer la place qui l'a lancé.

Pour faire face à la défaillance, les environnements d'agents mobiles offrent un mécanisme de point de reprise [15]. Les points de reprise sont conservés sur disque, support supposé fiable. Ils peuvent ainsi être utilisés ultérieurement pour restaurer l'agent en cas de défaillance. Cependant, lors de la restauration de l'agent il faut veiller à ne pas avoir deux agents actifs en même temps.

1.13. Plate-forme et Standardisation

La plupart des plates-formes pour agents mobiles ont introduit un ensemble de concepts similaires qui ont servi d'apport dans la standardisation des intergiciels agents.

Elles sont constituées de plusieurs entités. Nous pouvons distinguer [12] :



Chapitre 01:L'approche agent mobile

- **Les agents** ils sont définis par un identifiant unique au sein de la plate-forme et d'un état contenant plus d'un agent.
- **Les places** C'est l'élément qui sera en charge d'exécuter le code de l'agent.

Une place peut contenir plus d'un agent. Une place définit un environnement d'exécution vers lequel un agent sera orienté pour gérer son exécution.

- **Les agences** Encore appelées «ports d'accueil» des agents, représentent les environnements où évoluent les agents. Une agence constitue le cœur de gestion de la plate-forme en assurant la délégation de l'exécution aux places qui la composent, l'administration, le contrôle, le transport effectif et la communication des agents grâce au système d'exploitation sous-jacent.
- **Les régions** Il s'agit du regroupement de plusieurs agences, pas nécessairement de même type, appartenant à un même domaine d'expertise, d'activité ou autre. Le but est de faciliter les activités d'administration. On peut associer par exemple une région à des politiques d'accès aux services.
- **Les services** On retrouvera un ensemble de services accessibles aux agents. Ils pourront être construits sur différents modèles, Client/serveur ou agents, et fourniront aux agents en visite les fonctionnalités recherchées.

1.14. Les normes

L'émergence des nombreuses plates-formes expérimentales a rendu nécessaire la proposition d'une harmonisation grâce à la standardisation des différents concepts communs pouvant être identifiés. Cette normalisation devrait permettre à terme de rendre compatibles les différents systèmes.

La technologie agent faisant l'objet de nombreux travaux, une intégration des concepts clés est apparue essentielle avec comme finalité la production de spécifications destinée à l'industrie. C'est l'objectif que s'est fixé la (Fondation for Intelligent Physical Agents, FIPA) que de promouvoir le développement et la spécification de la technologie agent [14]. De même (Object Management Group, OMG) a étendu le champ de ses travaux initialement fondé sur le paradigme objet pour intégrer la technologie agent mobile en vue de produire des spécifications, Depuis peu, les deux organismes font des efforts conjugués pour fusionner les deux approches.

On peut trouver à l'heure actuelle deux normes principales [10] :

- **FIPA (Fondation for intelligent physical Agents)**

En revanche, la communauté d'origine de FIPA étant celle des systèmes multi agents, plus proche de l'intelligence artificielle, FIPA s'intéresse plus particulièrement à l'interopérabilité des agents intelligents (les efforts sont placés au niveau du langage, des protocoles et des infrastructures de communication), elle va se situer à un niveau plus élevé c'est à dire le niveau applicatif, en décrivant les éléments nécessaires à la réalisation d'une application et principalement en détaillant la communication entre les agents. Le but est de décrire un ACL (Agents Communication Language) et des protocoles de négociation permettant ainsi de définir parfaitement les interactions entre les agents.



Chapitre 01:L'approche agent mobile

• MASIF (Mobile Agent System Interoperability Facility)

La norme MASIF a été spécifiée par l'Object Management Group (OMG) qui se préoccupe généralement de l'hétérogénéité entre les systèmes, l'OMG s'intéresse à l'interopérabilité des agents mobiles à travers sa spécification appelée MAF.

Dans cette optique, le but, dans la norme MASIF, est de décrire les notions élémentaires permettant l'échange des agents entre différentes plates-formes. Pour ce faire, elle standardise la manière de gérer le code des agents, leur identification, la migration et l'adressage local.

1.14.1 But de la liaison

La raison de créer l'accord de liaison OMG-FIPA est d'encourager des normes de technologie d'agent à se développer compatible avec des normes de technologie d'objet et à la nouvelle coordination entre OMG et le travail lié du FIPA vers des normes de technologie d'agent [13].

1.15. Implémentations existantes

A l'heure actuelle, il existe beaucoup d'implémentation, incompatibles entre elles. Certaines de ces plates-formes existantes à nos jours sont présentées ci-dessous :

- **LIME** : le premier système que nous étudions est l'intergiciel basé sur Java nommé LIME (Linda In Mobile Environnement) qui propose une couche de coordination pour les agents en réutilisant le modèle Linda. Ce n'est pas à proprement parler une plate-forme car elle ne prend pas en charge les différents éléments décrits par les normes. Elle se focalise plus sur la coordination qui peut être exploitée pour construire des applications réparties.
- **Aglets**: utilise la notion de référence distante (AgletProxy), le passage de messages (multicast, broadcast), développé par IBM . Il offre en outre une certaine notion de sécurité en limitant les ressources allouées aux aglets.
- **JADE** : (java Agent Développement Framework), est un intergiciel construit sur java afin de permettre une programmation multi-agents simplifiée en prenant en compte la norme FIPA.
- **TACOMA** : le projet TACOMA (Troms And Cornell Moving Agents), de Norway & Cornell University, a été développé dans le but d'offrir différentes abstractions de haut niveau pour les agents. L'accent est mis sur un découplage, entre le langage et le niveau agent, semblable à celui introduit dans les normes à travers la notion de place.
- **PLANGENT** est une plate-forme basée sur le langage Java s'intéressant à l'adaptation des agents durant leurs déplacements au sein des environnements dynamiques. Son but est de proposer aux agents des mécanismes pour modifier les objectifs intermédiaires nécessaires à la réalisation de leur tâche finale[16].

1.16. Avantages de la technologie Agent Mobile

- Diminution de l'utilisation du réseau (dans les applications nécessitant beaucoup d'échanges avec le serveur)
- Les connexions avec le serveur sont locales donc l'application est moins sensible aux pertes de liens physiques
- Meilleure expressivité dans la conception d'applications réparties.



Chapitre 01:L'approche agent mobile

Conclusion

Nous avons présenté dans ce chapitre une vision générale sur le besoin crucial d'intégrer les agents mobiles en conséquence de l'incapacité du paradigme client/serveur et les remèdes suscités pour dépasser les inconvénients de ce paradigme. Ces agents mobiles sont organisés en des sociétés d'agents qui peuvent porter résolution aux problèmes de l'IA classique et qui interagissent, communiquent et coopèrent entre eux pour accomplir une tâche bien déterminée. On parle dorénavant d'une nouvelle technologie d'agents mobiles.

Par la suite ; nous avons décrit brièvement cette nouvelle technologie d'agents mobiles. Les agents mobiles sont des entités logicielles autonomes qui peuvent suspendre leur exécution sur une machine et migrer avec leur code, variables et états vers une autre machine où ils reprennent leur exécution.

Dans le deuxième chapitre de ce mémoire, nous allons présenter une étude détaillée sur « l'Internet des Objets ».

CHAPITRE 02

Internet des Objets



Chapitre02: Internet des Objets

INTRODUCTION

L'internet des objets est une notion complexe. Entre les puces RFID, les solutions de nommage ou middlewares, l'internet des objets est composé de nombreux éléments complémentaires ayant chacun leurs propres spécificités. Pour mieux appréhender la notion, nous proposons de définir les grands principes de l'IoT tels qu'ils apparaissent aujourd'hui à la lecture des recherches et des rapports réalisés sur la question[17].

Pourquoi cette évolution?

Selon Trend watching, ce sera pour mieux prendre soin d'eux-mêmes que les consommateurs adopteront les objets connectés. Les innovations vouées à s'implanter seront celles qui apporteront de nouvelles façons de répondre aux besoins fondamentaux, tels que la santé physique, le bien-être psychologique, la sécurité et les liens avec les personnes chères.

Cinq tendances accélèrent l'explosion des objets connectés.

- Des puces de connectivité sans fil miniaturisées, moins chères et plus efficaces qui permettent de connecter toute une variété d'objets, de la brosse à dents au cadenas pour vélo.
- Le stockage en ligne des données permet d'héberger la grande quantité d'information créée par les objets connectés.
- Les téléphones intelligents sont omniprésents et bénéficient d'une géolocalisation très précise. Il s'agit d'une condition nécessaire pour certaines interactions avec ces objets.
- Les capteurs sont désormais capables de mesurer tout (ou presque), de la luminosité au bruit et à la composition de l'air.
- La révolution du financement participatif a généré une marée d'innovations. Tout est possible, le consommateur n'a qu'à choisir.

Les objets connectés fonctionnent généralement avec les technologies du Near Field Communication (NFC) ou celle de son récent concurrent, la technologie iBeacon d'Apple. Il s'agit d'une nouvelle catégorie de transmetteurs qui peuvent reconnaître et envoyer des notifications aux périphériques SOA 7 et Android à proximité. Grâce à cette technologie, on peut, par exemple, localiser un individu passant devant un magasin et lui envoyer une alerte personnalisée (voir l'exemple d'Estimote). iBeacon rendra aussi possibles les paiements par téléphone intelligent simplement en se trouvant à proximité.



Chapitre02: Internet des Objets

1. Définition de l'internet des objets

Le terme d'Internet des Objets ne possède pas encore de définition officielle et partagée, qui s'explique par le fait que l'expression est encore jeune et que le concept est encore en train de se construire.

Cette définition illustre le concept d'Internet des Objets

« L'Internet des Objets est un réseau de réseaux qui permet, via des systèmes d'identification électronique 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, sans discontinuité entre les mondes physiques et virtuels, les données s'y rattachant. » [17].

2. Historique de l'internet des objets :

L'infographie proposée par Dassault remonte l'histoire des objets connectés jusqu'aux années 1800, période de l'invention du télégraphe, le premier appareil électronique de radio-communication. En 1926, c'est la première fois où Nikola Tesla imagine un monde où les habitants seraient reliés par des installations de communication sans fil. A partir de 1989, Tim Berners-Lee pose les bases du World-Wide-Web, l'internet universel que nous connaissons aujourd'hui.

Les premiers objets connectés n'apparaissent que dans les années 1990. Il s'agit de grille-pain, machines à café ou autres objets du quotidien. En 2000, le fabricant coréen LG est le premier industriel à parler sérieusement d'un appareil électroménager relié à internet, Les années 2000 verront les premières expérimentations d'appareils connectés à Internet. Ils l'utilisent notamment pour consulter des informations de manière automatique, notamment Ambient Orb vers 2002. C'est bien avant qu'IPSO ne propose le concept d'adresses IP (2008) ou que l'organisation des Nations Unies ne mentionne l'internet des objets dans un rapport sur les télécommunications internationales.

Depuis 2011, l'IPv6 offre de nouvelles possibilités pour les objets connectés qui disposent de nouvelles plages d'adresses IP disponibles et attribuables. En 2013, Intel lançait l'Internet of Things Solutions Group et nous créions ce site en janvier Dassault Systèmes recommande quelques articles sur l'internet des objets écrits au cours de l'année 2013, dont ceux des magazines en ligne américains Wired, Venture Beat ou CNBC, des sources qui nous inspirent également au quotidien.

A la fin de l'année 2012, il y avait environ 8,7 milliards d'objets connectés dans le monde. Cisco estime que ce nombre atteindra sans mal les 50 milliards d'objets connectés en 2020. L'IoT mise sur le chiffre de 80 milliards ! En 2020, nous seront là pour vous en parler, quel qu'en soit leur nombre.



3. Les capteurs, composants majeurs de l'Internet des objets

L'essor des capteurs suscite un grand intérêt aussi bien dans la communauté de la recherche que dans le monde industriel. Les apports attendus de cette branche active, à la croisée de l'électronique, du réseau et de l'informatique, permettent d'envisager une perception améliorée d'un milieu. Cette technologie permet une évaluation chiffrée de l'environnement direct, des informations localisées et des traitements informatiques. L'ensemble des mesures doit être mis à disposition de l'utilisateur, et lui être transmis, ou du moins être accessible à distance, offrant alors une véritable valeur ajoutée. Pour y parvenir, ces objets de petite taille, à coût très réduit, pourront s'organiser entre eux et, pour peu qu'ils s'astreignent au respect de standards clairs et robustes, fournir l'accès le moins contraignant possible à toute personne/service désireux de l'utiliser[18].

3.1. Les capteurs

Les capteurs sont des appareils qui possèdent des capacités de mesures, voir d'actions, sur leur environnement. Ils peuvent capter des informations liées à la température, au taux d'humidité, à la luminosité ambiante, à la détection de présences, à la présence de gaz, de polluants ou encore à la géolocalisation. Tout cela fait partie des informations les plus couramment collectées par les capteurs. Selon les capteurs, la pression atmosphérique, l'intensité lumineuse ou la pression sonores peuvent aussi être quantifiés. Les spécificités nouvelles de ces capteurs résident dans leur taille et leur coût réduit, tout en étant dotés des capacités de traitements de l'information, et des possibilités de transmission sans fil.

La nature de l'objet "intelligent" dispose de possibilités de calcul et de transmission des données. L'échange d'informations, voir l'interaction entre différents objets est envisageable, et ce même sans intervention des utilisateurs est la base d'une certaine forme d'intelligence. En fait plus le nombre de capteurs est grand plus "l'intelligence" du système augmente comme un réseau de neurones. L'objet va "capter", mesurer une caractéristique physique de son environnement, éventuellement appliquer un traitement informatisé, et fournir le résultat à d'autres objets comme des ordinateurs[19].



Figure 2.1: Panorama des capteurs utilisés dans le laboratoire LIGM



Chapitre02: Internet des Objets

Les outils présentés figure 2.1 sont principalement à destination de la recherche, mais leurs fonctionnalités opérationnelles (bouton, led, capteurs de température et lumière, buzzer ...) suffisent pour tester des architectures réalistes.

3.2. Les réseaux de capteurs

Aujourd'hui, les réseaux sans fil sont de plus en plus populaires du fait de leur Facilité de déploiement. Ces réseaux jouent un rôle primordial au sein des réseaux Informatiques. Ils offrent des solutions ouvertes pour fournir la mobilité ainsi que des Services essentiels là où l'installation des infrastructures n'est pas possible.

Généralité

Un réseau de capteurs peut être vu comme un réseau de microsystemes autonomes disséminés dans un espace donné et communiquant entre eux via une liaison sans fil. le lieu où agissent les capteurs est nommé un champ de captage. Ce qui est intéressant dans les réseaux de capteurs, c'est l'aspect autonome car l'intelligence des capteurs permet de décentraliser l'intelligence du réseau. C'est à dire que tout le réseau ne dépendra pas de quelques capteurs.

Architecture

Un réseau de capteurs sans fils (En anglais : Wireless Sensor Networks (WSN)) est composé de capteurs ayants des capacités de détections et de mesure d'évènements, est munis d'éléments de communication qui donne à un administrateur la capacité de manipuler, observer et réagir aux évènements et aux phénomènes dans un environnement spécifique ,dit champ de captage. L'administrateur est une entité civile gouvernementale, commerciale ou industrielle.

Il y a quatre composants de base dans un réseau de capteurs: un ensemble de capteurs distribués ou localisés ; un réseau d'interconnexion (habituellement, mais pas toujours, sans fil) ; Un point central appelé puits (Sink) pour le regroupement et le traitement de l'information et ; un ensemble de ressources

informatiques déployées sur le Sink pour traiter les données, l'évolution des évènements, le statut requêtes et l'exploration de donnée(data mining) [19].

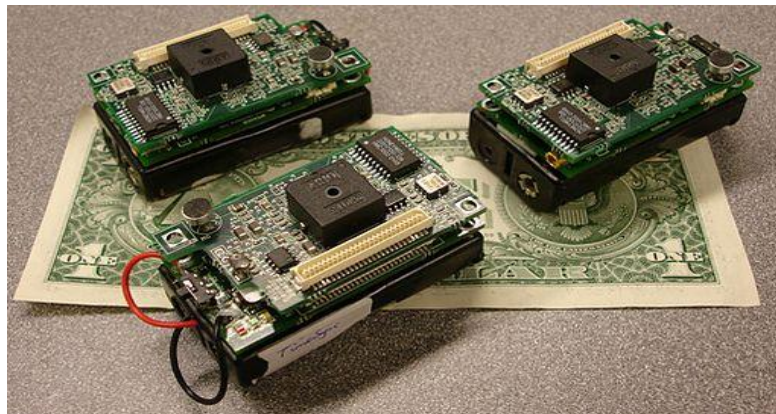


Figure 2.2: Capteurs sur un billet de 1 dollar utilisé comme échelle ce qui montre qu'ils sont très petits.



Chapitre02: Internet des Objets

3.3. L'Énergie

L'énergie représente une contrainte forte dans la majorité des applications à base de réseaux de capteurs sans fil : Chaque nœud capteur fonctionne avec une batterie, généralement, non rechargeable et à capacité limitée étant donné sa petite taille. Dans la majorité des cas, ces nœuds capteurs sont déployés dans des zones hostiles ou difficiles d'accès et il est très peu probable qu'ils soient récupérables. Aussi, vu leur nombre très grand (des milliers) on ne peut pas imaginer (faute de temps et coût) de s'occuper de chaque nœud capteur un à un. Vu ce constat (impossibilité de remplacer un nœud hors service pour problème d'énergie ou problème de fonctionnement), tout programme dédié à fonctionner sur un nœud capteur doit prendre en considération ce problème de gestion d'énergie.

Surtout en temps de transmission du fait qu'une communication sans fil est une des opérations les plus coûteuses parmi celles dont est chargé le nœud capteur[20].

3.4. Puissance de stockage et de traitement

La puissance de stockage et de traitement est relativement faible. Par exemple, les nœuds capteurs de type "mote" sont composés d'un microcontrôleur 8-bits 4MHz, 40Koctets de mémoire et d'une radio d'un débit d'environ 10 kbps. Cela reste vrai même pour les nœuds de moyenne gamme, comme les "UCLA/ROCKWELL'S WINS", qui ont un processeur String ARM 1100 avec une mémoire flash de 1Moctets, une mémoire RAM de 128 K octets et une radio dont le débit est 100 Kbps[20].

4. Les services de l'Internet et l'IoT

Dès lors que les différents nœuds communiquent, les informations dont ils ont la charge, ou les actions qu'ils peuvent accomplir, deviennent accessibles. Cependant, l'accès à celles-ci n'est pas forcément pratique. Dans quelle mesure son usage est-il connu, décrit, et selon quel mode opératoire? Qui peut y accéder, et comment? Dans un objectif d'utilisation simplifiée, voire automatique, ou même pervasive, quelles méthodes sont envisageables pour franchir le pas entre la possibilité d'un accès à une ressource offerte par le système, et son utilisation tangible et pragmatique?

Un panorama proposé des systèmes qui permettent de décrire les services disponibles dans un environnement ouvert et diversifié tel que celui que peut rencontrer un utilisateur, encore des mécanismes proposent de faire interagir ces services entre eux[21].

4.1. L'approche services SOA

SOA (Services Oriented Architecture) est une approche de l'Internet des PC dans laquelle l'ensemble des informations, des traitements, et des ressources du système d'information sont présentées sous la forme de services. Il ne s'agit pas à proprement parler d'une norme, mais plutôt d'une architecture, d'une construction assemblant les technologies existantes. La finalité de cette démarche de conception est de parvenir à l'interopérabilité entre les outils et solutions employées, qui doivent être normalisés et les plus répandus possible (dans le cas contraire, cette inter-opérabilité serait plus difficile à assurer)[21].



Chapitre02: Internet des Objets

Plusieurs organismes, comme l'OASIS, se consacrent à la production et description de normes garantissant une standardisation des constituants SOA. L'objectif de cette démarche est de rendre possible une composition, un assemblage de ces services afin de construire de nouvelles applications. Plutôt que d'avoir des applications monolithiques, et d'ensuite concevoir des passerelles entre ces différents traitements (sous forme d'export, par exemple, ou tout autre mode de conversion), la démarche SOA opte pour l'utilisation de briques logicielles, accessibles, ouvertes, décrites, offrant des services atomiques. L'architecture ainsi réalisée est alors découplée, c'est-à-dire que les différents éléments échangent de l'information tout en étant très peu dépendants les uns des autres, uniquement liés par le contrat stipulant la façon de communiquer. On obtient alors une solution faiblement couplée, chaque élément pouvant évoluer sans remettre l'édifice en cause, pour peu qu'il continue à respecter la même interface visible[22].

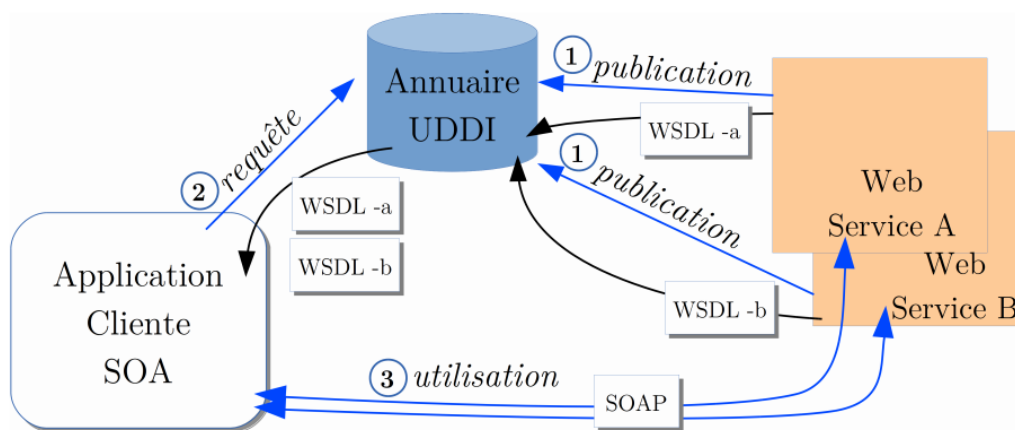


Figure 2.3: L'Architecture Orientée Service (SOA).

Autant cette démarche a du sens dans la construction d'un service qui sera offert à de très nombreux utilisateurs, autant dans le cadre de l'Internet des objets les rapports évoluent, et l'organisation est amenée à être repensée. Dans l'Internet des objets, le circuit concepteur/consommateur est réduit, et parfois ces rôles sont tenus par une même et unique personne.

4.2. Les services web

Les services web sont aux applications clientes ce que le Web est aux utilisateurs. Ici, on utilise ce même protocole applicatif, mais dans l'idée d'offrir des informations adaptées à leur utilisation par d'autres programmes, et non plus par un être humain. Les besoins en termes de présentation de l'information et de sa granularité sont spécifiques. Souvent, une application a besoin d'une simple donnée, afin de la traiter (un horaire, un prix, un nom, une url ...). La présentation qui lui importe concerne plutôt le typage de la donnée et son encodage, et non un format de mise en page[23].



Chapitre02: Internet des Objets

Les services web (Web-Service) sont la déclinaison principale de l'architecture SOA utilisant le protocole HTTP. Si un noeud doit offrir un tel service, pourquoi ne pas utiliser un protocole répandu, bien connu et robuste? Habituellement, le World Wild Web, permet l'accès d'un utilisateur humain au contenu d'une ressource stockée sur une machine distante. HTTP décrit l'établissement de la connexion TCP vers cette ressource, décline un ensemble de paramètres pour spécifier la ressource voulue (l'URI), et détermine la façon dont le serveur va répondre (entête HTTP) avant que celui-ci ne ferme la connexion (sauf dans le cas du keep-alive de la version 1.1 du protocole). Le contenu de l'échange, sous forme de texte, est souvent encapsulé dans un format html afin d'agrémenter sa présentation pour le navigateur de l'internaute [24].

4.3. L'approche REST

Representational State Transfert (REST) est un style d'architecture permettant de construire des applications (Web, Intranet, Web Service). Il s'agit d'un ensemble de conventions et de bonnes pratiques à respecter et non d'une technologie à part entière. L'architecture REST utilise les spécifications originelles du protocole HTTP, plutôt que de réinventer une surcouche (comme le font SOAP ou XML-RPC par exemple)[25].

Règle N°1 : l'URI comme identifiant des ressources.

Règle N°2 : les verbes HTTP comme identifiant des opérations.

Règle N°3 : les réponses HTTP comme représentation des ressources.

Règle N°4 : les liens comme relation entre ressources.

Règle N°5 : un paramètre comme jeton d'authentification.

Dans le cadre des capteurs et des environnements très contraints en général, il semble pertinent de disqualifier SOAP, notamment par sa lourdeur, au regard du service qu'il est censé rendre. Cette critique qui a été conçue pour les environnements réseaux plus puissants de l'Internet s'impose d'autant plus pour les réseaux de capteurs. La plupart des opérations courantes peuvent être réalisées grâce à HTTP, plus simplement, de façon plus légère et concise. La concision et la légèreté des échanges sont précisément ce que les capteurs requièrent, au regard de leurs contraintes énergétiques[26].

4.4. La chorégraphie et l'orchestration de services

La disponibilité de services web chargés de la réalisation de tâches atomiques, disséminés sur le réseau, flexibles et autonomes permet d'en imaginer des combinaisons, dans le but de fabriquer un processus complet. La construction d'une telle combinaison s'appréhende de deux façons différentes. D'un côté la conception d'un algorithme sous le contrôle d'un point central, chargé de l'enchaînement des appels de services et de la gestion des résultats retournés, ou orchestration de services. A contrario, une approche dans laquelle chaque service interagit avec les autres sans être placé sous la responsabilité d'un noeud principal est dénommée une chorégraphie de services[27].



Chapitre02: Internet des Objets

La vue orchestrée de la conception d'une application reste assez conforme aux démarches des programmeurs, dans laquelle un programme principal est chargé du "flux de contrôle", et multiplie les appels à des ressources (qui sont ici les services web). BPEL est le langage le plus couramment utilisé pour modéliser des orchestrations[28].

La chorégraphie quant à elle met plus l'accent sur les relations entre les différents services, c'est-à-dire les causes, les dépendances et les exclusions. elle ne décrit pas d'effets internes (alors que l'orchestration au contraire se concentre sur le raisonnement interne du noeud principal) mais plutôt les réactions de chacun des services soumis à des stimuli échangés.

décrivent le déroulement de différents services en parallèle, chacun recevant des messages d'un autre, réagissant en progressant dans son propre flux de contrôle, puis déclenchant une action à son tour[29].

L'accent sur ce segment réside dans l'attention l'utilisation de la chorégraphie dans le cadre de SOAP et de REST, et compare leurs points forts et faibles face aux fonctionnement de la chorégraphie. REST fournit un couplage faible, et la clarté de la définition des opérations voulues qui apparaissent dans le message, tandis que SOAP permet de décrire la complexité de certaines opérations et propose un suivi des échanges, intéressant notamment pour déboguer la cinématique du dialogue [30].



Figure 2.4: Quelques objets intelligents mis en oeuvre dans l'Internet des objets.

Nous pensons que l'approche chorégraphiée est adaptée à l'Internet des objets . Elle permet à la fois d'alléger le recours aux ressources des réseaux WSN qui participent à l'IoT. La chorégraphie respecte les contraintes énergétiques, car les interactions directes entre noeuds sollicitent moins le réseau qu'une remontée des données à un point central. De plus, les capacités de traitement ajoutées aux objets (certes limitées, mais bien présentes) permettent l'implémentation des algorithmes décrivant la tâche à effectuer sur l'objet lui-même[31].



Chapitre02: Internet des Objets

5. Les protocoles à l'oeuvre dans l'Internet des objets

L'Internet des objets est à la croisée d'un ensemble de technologies. C'est la démocratisation et l'accès à de multiples sources de données et traitements par des moyens de plus en plus transparents (sans fil, automatisé, auto-négocié) qui permettent d'en imaginer la forme. Les capteurs ont un rôle non négligeable à jouer dans ce domaine, car ils sont les objets qui vont se relier, grâce à l'Internet, à l'ensemble des outils dont nous sommes familiers. Cependant, l'Internet des objets ne se limite pas à ceux-ci, et intègre une plus vaste gamme d'éléments, aussi bien provenant de la domotique, que de l'Internet de PC traditionnel. Il est donc nécessaire d'à la fois organiser le réseau et la connexion à ce réseau, mais aussi proposer un accès efficace et exhaustif aux services mis à disposition[32].

5.1. DPWS (Devices pour les services Web)

Cette section présente le profil Devices pour les services Web (DPWS). DPWS est aussi parfois appelé Web Services sur Devices (WSD). Un PC ou un autre périphérique peuvent détecter les périphériques DPWS activés sur un réseau, puis découvrir et invoquer la fonctionnalité de service Web chaque dispositif fournit.

En termes de services Web, un profil est un ensemble de lignes directrices pour l'utilisation de technologies de services Web dans un but donné. Les normes de services Web permettent implémenter de choisir parmi une variété de messages représentations, encodages de texte, les protocoles de transport et d'autres options, dont certaines ne sont pas interopérables. En limitant ces décisions, les profils assurent que les implémentations conformes au profil va bien travailler ensemble.

Le DPWS offre les fonctionnalités suivantes entre des appareils compatibles telle que Les dispositifs Découvrir DPWS compatibles sur le réseau et les services qu'ils offrent, et l'envoi de messages à des dispositifs DPWS-capables et réponses recevant, et la description d'un service Web en fournissant un fichier WSDL.

Ces fonctionnalité permettent de :

Communiquer entre les périphériques.

Interagir avec un service en utilisant sa description.

Abonner et recevoir des événements à partir d'un service Web.

Les appareils peuvent être des clients DPWS (invoquant des services sur les appareils), les serveurs (fournissant des services), ou les deux. DPWS pour le Micro Framework .NET prend en charge les périphériques soit dans le rôle ou les deux simultanément. Ces deux ensembles de fonctionnalités sont fournies DLL séparées et ne sont pas dépendantes les unes des autres.

DPWS est basé sur les normes existantes de services Web, y compris XML, WSDL, XML Schéma, SOAP, MMD et HTTP. DPWS pour le .NET Micro Framework intègre également cette fonctionnalité, mais pas tout cela est exposé dans l'API. Par exemple, il y a des classes pour lire et écrire des documents XML afin que les applications et les implémentations de services peuvent analyser Web messages services et de construire des réponses pour eux[33].

5.2. Problématique spécifique aux réseaux de capteurs

La plus importante concerne l'énergie, et toutes les autres en découlent. Les choix concernent principalement une limitation de la capacité de traitement, un espace mémoire très limité, une couche réseau à débit très faible et une taille de charge utile limitée. Ces caractéristiques vont



Chapitre02: Internet des Objets

peser de façon non négligeable sur le portage d'une architecture à base de services sur ces équipements. En effet, les architectures SOA ont été conçues pour des réseaux dans lesquels les débits sont importants, et qui utilisent des protocoles réseau dont les trames supportent des charges utiles conséquentes, et dont les équipements disposent de confortables ressources en mémoire, en capacité de traitement, et d'une énergie illimitée.

L'approche services est de plus en plus acceptée comme solution efficace pour une utilisation plus "applicative des réseaux de capteurs". Le portage de DPWS sur les réseaux de capteurs a fait l'objet du projet européen SODA. Comme dans sa version générique, les capteurs se découvrent, présentent leurs services, s'enregistrent éventuellement les uns auprès des autres, puis s'échangent de l'information. Concevoir des applications mettant en oeuvre ces capteurs devient alors possible.

La simple implémentation d'un serveur HTTP pose problème, car l'utilisation de TCP sur des architectures si limitées en mémoire est quasi-proscrite. La mise à disposition d'un service web à contenu dynamique transporté dans des messages XML de type SOAP telle qu'elle est organisée dans le monde de l'Internet des PCs semble grandement compromise au vu des caractéristiques des objets chargés d'héberger ce service. Il faut donc repenser la façon dont celui-ci est construit[34].

5.3. Services dans un environnement de capteurs

Dans le domaine assez récent des réseaux de capteurs, la nécessité de consulter les valeurs mesurées, et ce le plus simplement possible, est un enjeu majeur pour leur utilisation. Les capteurs sont capables de mesurer certaines caractéristiques de leur environnement, et peuvent ensuite transmettre ces informations au puits, chargé de les collecter. L'inversion de cette démarche permet d'imaginer un usage dans lequel c'est l'extérieur qui va interroger le capteur afin de récupérer la mesure.

Les Capteurs et actuateurs distribués dans plusieurs maisons et bureaux peuvent augmenter le confort dans ces environnements: le chauffage peut s'adapter à la météo, l'éclairage suivre l'horaire et la position du soleil, des incidents domestiques peuvent être évités avec des alarmes et beaucoup d'énergie pourrait être économisée.

Mais des environnements intelligents peuvent améliorer l'automatisation en milieu industriel avec un déploiement massif de tags RFID associés aux différentes étapes de la production.

La ville intelligente est un exemple d'environnement intelligent[35].

5.4. CoAP (Application Protocol Constrained)

Le protocole d'application Constrained (CoAP) est un protocole spécialisé de transfert de bande pour une utilisation avec des nœuds contraints et réseaux contraints à l'Internet des objets.

Le protocole est conçu pour les applications (M2M)-machine à machine tels que l'énergie intelligente et l'automatisation des bâtiments[36].

-Les ports et verbes HTTP sont compressés en un code défini au préalable, par la norme.



Chapitre02: Internet des Objets

-La passerelle chargée de la communication avec l'extérieur se réfère à ce code court pour offrir l'illusion du comportement habituel du protocole réel.

- Le système d'exploitation libre pour WSN, implémente le protocole. Avec CoAP, les interactions entre services web de l'Internet des PC et de l'Internet des objets deviennent bien plus simples à réaliser, une passerelle applicative assez légère (correspondance entre les commandes REST et CoAP) se charge de l'adaptation d'un monde à l'autre.

6. Quelles solutions architecturales pour l'IoT?

La création d'applications clé en main se heurte à la grande diversité des parties prenantes de l'Internet des objets, à laquelle s'ajoute le poids de l'existant. Les protocoles applicatifs de l'Internet ne sont pas adaptés, à la puissance très limitée des objets (du moins la majorité d'entre eux, dans le cas des capteurs). Pourtant, l'Internet des objets ne peut émerger qu'à travers la nécessaire universalité de sa proposition architecturale, puisque la population d'objets prévue devrait dépasser de plusieurs facteurs l'actuel nombre de machines connectées au réseau des réseaux[28].

6.1. Les systèmes d'exploitation spécialisés

Avec l'explosion des objets connectés, les systèmes d'exploitation embarqués s'adaptent. Consommation, stabilité, évolutivité, solutions open source ou propriétaires, les choix s'étoffent. Retour sur le rendez-vous dédié à l'IoT. « Des érables aux solutions de la ville connectée ... la prochaine révolution est celle des objets connectés », dédié à l'IoT[37].

Développer une solution pour un OS précis est trop limitant au regard de la diversité du matériel impliqué. Cette restriction est l'un des principaux écueils, à notre sens, à l'essor de l'Internet de objets. Nous préférons nous orienter vers d'autres solutions qui fournissent une vision plus uniforme des objets, propice à l'évolution du parc mis en oeuvre, ouvert à une large gamme de matériel.

6.2. Les machines virtuelles

Une machine virtuelle est un conteneur logiciel totalement isolé capable d'exécuter son système d'exploitation et ses applications comme s'il s'agissait d'un véritable ordinateur. Les machines virtuelles sont des solutions intéressantes de contournement de l'hétérogénéité. Elles masquent les différences de matériel en offrant un profil unifié et une interface normalisée. Ensuite, dans certains cas comme Java, les programmes créés peuvent prendre la forme de byte code dont la taille est souvent limitée, favorisant alors le déploiement par le réseau (légèreté et universalité).

Le résultat est fonctionnel, permet l'exécution de programmes WORE (écrits une fois, exécutables partout). Cette universalité est propice à l'écriture d'applications Internet des objets. Toutefois, les besoins applicatifs sur chaque noeud devraient pouvoir être décrits au moyen de langages plus simples et adaptés que Java. Darjeeling n'offre qu'un sous-ensemble des fonctionnalités du langage, impliquant une connaissance précise du programmeur sur les limites de la machine virtuelle par rapport à la norme du langage[38].



Chapitre02: Internet des Objets

6.3. Les middlewares

On appelle middleware (ou logiciel médiateur en français ou encore intergiciel), littéralement « élément du milieu », l'ensemble des couches réseau et services logiciel, qui permettent le dialogue entre différents composants d'une application répartie. Ce dialogue se base sur un protocole applicatif commun, défini par l'API du middleware. Le Gartner Group définit le middleware comme une interface de communication universelle entre processus. Il représente véritablement la clef de voûte de toute application client-serveur.

L'objectif principal du middleware est d'unifier, pour les applications, l'accès et la manipulation de l'ensemble des services disponibles sur le réseau, afin de rendre l'utilisation de ces derniers presque transparente[39].

- Les services des middlewares

Un middleware est susceptible de rendre les services suivants:

- Conversion : service utilisé pour la communication entre machines mettant en oeuvre des formats de données différents.
- Adressage : permet d'identifier la machine serveur sur laquelle est localisé le service demandé afin d'en déduire le chemin d'accès. Dans la mesure du possible.
- Sécurité : permet de garantir la confidentialité et la sécurité des données à l'aide de mécanismes d'authentification et de cryptage des informations.
- Communication : permet la transmission des messages entre les deux systèmes alternation. Ce service doit gérer la connexion au serveur. La préparation de l'exécution des requêtes, la récupération des résultats et le dé - connexion de l'utilisateur[40].

6.4. Les enjeux et proposition des solution de l'IoT

L'Internet des objets est à la croisée de plusieurs domaines et suscite l'intérêt de la communauté académique comme des industriels. Il s'agit de ne pas passer à coté des opportunités associées à l'essor de l'IoT.

Le succès massif de certaines entreprises telles que Google suscite, chez les opérateurs et les fournisseurs d'accès, le sentiment que de telles réussites devraient se traduire par une participation au financement des infrastructures. Ils tendent de ce fait à remettre en cause ce qui constituait jusque-là l'un des principes fondateurs de l'internet : à savoir sa neutralité et le traitement indifférencié des données ou des contenus y circulant. Il faut s'attendre à ce que la question se pose rapidement en matière d'IoT. En effet, ce dernier ne peut que renforcer l'écart déjà existant entre les modèles économiques à la base des infrastructures et ceux des utilisateurs ou secteurs applicatifs, qu'il s'agisse d'industriels intégrant l'IoT dans leurs processus de production et leur logistique, ou de fournisseurs de services grand public[25].

6.5.L'intégration des objets dans l'Internet

Il s'agit donc de reproduire sur ces objets hétéroclites un comportement le plus proche possible de celui attendu d'un serveur. Les objets devront être programmables, afin de livrer



Chapitre02: Internet des Objets

un service évolutif et adapté. Les premiers travaux ayant retenus notre attention sont ceux de présente des solutions pour l'Internet des objets bâties sur des serveurs web hébergés sur chaque objet. Cette implémentation valide le concept, et utilise des méthodes de pré-calcul des paquets permettant de composer ceux-ci à la volée, avec du contenu dynamique, tout en allégeant le traitement nécessaire à la fabrication des messages, le tout pour une empreinte mémoire réduite[41].

Enfin, propose un serveur très léger, programmable, interagissant avec les objets, le thin server. Ce serveur, ou du moins chacune des instances représentant l'objet distant, peut ensuite être rapidement reprogrammé par l'utilisateur via à un moteur de composition. Actinium recense tout un ensemble de comportements que les objets peuvent adopter, et en permet l'activation et l'interaction.



Chapitre02: Internet des Objets

Conclusion

L'internet des objets suscitera des questions, qui concerneront directement la sécurité des biens et des personnes. Par exemple, certaines applications peuvent être étroitement liées à des infrastructures stratégiques telles que la fourniture d'eau et d'électricité, la surveillance de ponts et bâtiments, et la gestion d'un centre de commerce que nous en allons concevoir notre application.

Nous avons évoqué Les enjeux de l'IoT et les solutions compatible.

Ensuit nous allons exposer au chapitre suivant la modélisation de l'IoT.

CHAPITRE 03

Modélisation



1. Introduction

L'Internet des objets (IoT) est un paradigme moderne. L'idée de base de ce concept est la présence omniprésente autour de nous d'une variété des choses ou des objets tels que Radio-Fréquence D'identification (RFID), les capteurs, les actionneurs, les téléphones mobiles, etc., qui, grâce à des schémas d'adressage uniques, sont capables d'interagir les uns avec les autres et coopérer avec leurs voisins pour atteindre des objectifs communs [41]. L'IoT est un environnement distribué caractérisé par plusieurs dispositifs hétérogènes, y compris les capteurs, les actionneurs, les appareils, les téléphones, les périphériques de stockage. IoT transforme objet physique en un réseau sans fil interconnecté avec ces dispositifs pour communiquer coopérer et partager les informations et de réagir à des changements dans l'environnement [42]. L'IoT est un système adaptable, il assure une gestion autonome et auto-configurer, cette capacité d'adaptation peut changer l'environnement distribué facilement. Les aspects du système multi-agents peuvent modéliser l'environnement IoT distribués, ils peuvent modéliser l'autonomie, l'adaptabilité et la coopération des systèmes, La proposition de Kortuem et ses collaborateurs adopte l'approche mobile pour IoT dans un environnement distribué, afin de faciliter et d'optimiser les tâches et les actions d' IoT. Les approches IoT est un système distribué. De nombreuses questions de recherche posées: (1) Comment l'architecture de système est distribuée? (2) Quels sont les modèles seront utilisés? (3) Comment représenter les objets intelligents et les capacités de l'intelligence distribuée? (4) Comment combiner une application collective cohérent ?

Nous cherchons à répondre à ces questions par une approche basée sur l'agent mobile dans un environnement distribué.

2. Les travaux similaires

Dans Leppnen, Teemu, et al. : Mobile agents-based smart objects for the internet of things les ressources d' IoT sont représentés par un agent [43]. Cella est supposé d'être en mesure de surveiller l'état de la ressource, de prendre des décisions au nom de la ressource, et de découvrir, demander et d'utiliser une aide extérieure si nécessaire. La liaison entre la ressource elle-même et son agent est organisée par l'intermédiaire d'un adaptateur sémantique (ou interface). Un tel adaptateur peut comprendre (si nécessaire) des capteurs avec sortie numérique, structural des données (ex-XML), composants d'adaptateur sémantiques

Chapitre03: Modélisation



(conversion d'une représentation sémantique), et des actionneurs. Selon cette vision, chaque composante pertinente d'un système ubiquitaire est devenir une ressource intelligente proactive et l'autogestion. La proposition de Kortuem et ses collaborateurs, l'architecture à base d'agents pour les objets intelligents, où le système hétérogénéité IoT est abstrait avec des couches. L'architecture du système permet d'une communication SOA aux détails sous-jacents abstraits, une composante de la gestion de la communication avec des systèmes externes, un module de découverte de ressources, des adaptateurs à des capteurs abstraits et actionneurs que les ressources du système et des composants enfin pour la gestion des contextes, la base de connaissances et de raisonnement[42]. La mise en oeuvre est à base de Java. Le modèle maître-esclave est utilisé avec des objets intelligents, où un coordonnateur gère l'ensemble des entités logicielles, en cours d'exécution sur d'autres objets intelligents. Le coordonnateur contrôle l'objet intelligent d'hébergement si le protocole de communication interne est le seul composant à communiquer avec d'autres objets intelligents ou les dispositifs du système, à travers le protocole de communication externe. Un cadre de logiciel interne fournit l'API pour les services atomiques et gère l'environnement d'exécution comme un cadre supplémentaire du logiciel interne. Dans l'article de Shelby, Z. la langue et la plate-forme présente une composition indépendante pour les agents mobiles basés sur les objets intelligents. Cette méthode est basée sur des standards ouverts pour la communication sur les réseaux disparates et pour le soutien de la collaboration sans modèles d'interaction spécifiques ou demi-fil. L'infrastructure de l'information est réalisée avec le cadre IETF Coré, permettant en outre des plates-formes de périphériques de ressources limitées pour les objets intelligents de devenir partie intégrante de t-il Web. L'architecture du système est plat et ne se limite pas à des modèles spécifiques d'interaction, de communication ou de programmation. la configuration du système centralisé ou la coordination de l'agent ne sont pas facilitées et nous ne sont pas applicables toute configuration de système ou d'un plan de travail spécifique avec les objets intelligents[44].

Au lieu de cela, ils exposent la composition d'agent modifiable dans le système comme une ressource commune. Cette méthode facilite dynamique inter-reliés plusieurs à de nombreuses communications, y compris les systèmes externes, malgré les orles de l'agent ou des objets intelligents. Les principes de conception REST et les interfaces unifiées sont utilisés pour la création de l'agent, la migration, la messagerie, le contrôle et l'exposition des ressources système sur le Web. Enfin, bien que les composants logiciels Java sont modulaires, portables



Chapitre03: Modélisation

et offrent des fonctionnalités orientées objet pour la programmation, des solutions à base de machine virtuelle Java peut être trop lourd pour la plupart des systèmes embarqués à ressources limitées. Dans l'article le livre [42] la présentation intelligente d'Environnement Hiérarchie (SETH) architecture pour le développement d'environnements intelligents utilisant des agents logiciels. Ils ont proposé de placer des agents logiciels entre les utilisateurs et l'environnement. Cette présentation l'intégration de l'IoT et réseau de capteurs sans fil avec des agents mobiles. Les avantages attendus comprennent: agent- composition de service adaptable en fonction est facilitée, la charge de calcul est distribuée, la localité peut être exploitée dans les ressources de communication et de système sont exposés sur le Web pour l'interaction homme-machine[39].

3. Environnement Distribué dans l'IoT

Atzori et Al résume l'architecture d'IoT, cette architecture est composée par cinq couche, objet abstraction, la gestion des services, la composition des services et des applications , discussion.

3.1.Objets abstraction

Les objets Abstraction concernent la couche d'abstraction d'objet. La capacité d'harmoniser l'accès aux différents dispositifs avec un langage et de procédure commun[43].

3.2. La gestion des services

Cette couche fournit les fonctions principales qui devraient être disponibles pour chaque objet et qui permettent de leur gestion dans le scénario d' IoT. Un ensemble de services de base comprend: découverte d'objet dynamique, surveillance de l'état et la configuration du service. A cette couche, certaines propositions comprennent un ensemble étendu de fonctionnalités liées à la gestion de la IoT et de gestion de verrouillage, ainsi que certaines fonctions sémantiques par exemple, la gestion du contexte). Cette couche permettre le déploiement à distance de nouveaux services lors de l'exécution, afin de satisfaire les besoins de l'application. Un référentiel de service est construit à cette couche afin de savoir le catalogue de services qui est associés à chaque objet dans le réseau. La couche supérieure peut alors composer des services complexes en se joignant à des services fournis à cette couche [43].



3.3. Service de composition

Ceci est une couche commune au sommet d'une architecture IoT. Il fournit les fonctionnalités pour la composition des services uniques offerts par des objets en réseau pour créer des applications spécifiques. Sur cette couche il n'y a aucune notion de dispositifs et les seuls actifs visibles sont des services. Un aperçu important dans le paysage de service est d'avoir un référentiel de toutes les instances de service actuellement connectés, qui sont exécutés dans le moment de l'exécution pour construire des services composés [43].

3.4. Application

La couche application concerne les applications d'un utilisateur final [43].

3.5. Discussion

Il y a beaucoup de recherche adressée à proposer agent mobile dans l' IoT, mais ils ne pas concentré à l'environnement distribué, ils se concentrent sur les objets, l'agent, agent mobile afin de faciliter la gestion des données de système. Ces recherches capturent les objets dans leur conception, ils ne prennent pas en considération l'environnement de ces objets.

Notre vision capturer l'environnement afin d'enrichir le cadre conceptuel, nous avons une architecture IoT composé par l'Environnement et ses objets d'abstraction, service de l'environnement et sa gestion des objets, service de l'environnement et sa composition des objets et de l'application.

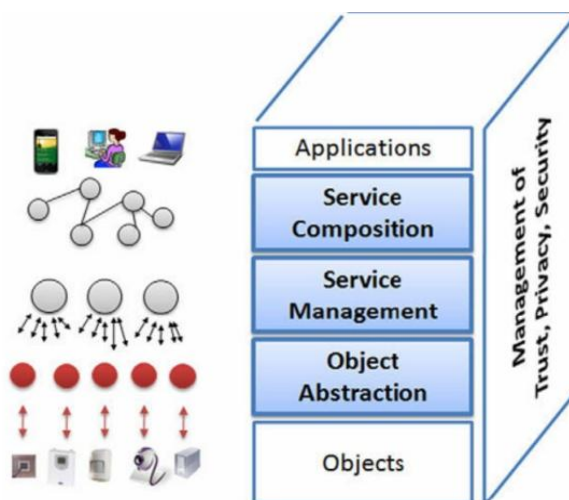


Fig3.1: IoT architecture basée sur des IoT SOA

Chapitre03: Modélisation



Notre proposition

Dans cette section, nous présentons notre approche pour développer un système IoT dans un environnement distribué.

Architecture IoT pour l'environnement distribué

Notre SOA axée sur l'environnement dans les différentes couches afin d'accomplir toutes les tâches dans un environnement distribué.

. Nous avons modélisé notre architecture IoT par 04 couches:

- Environnement et ses objets: Cette couche présente l'environnement et ses objets en monde réel.
- Environnement et ses objets abstraction: Dans cette couche on fait l'abstraction de l'environnement et ses objets, l'abstraction de l'environnement correspondant au type d'environnement, nous faisons l'abstraction du matériel et des logiciels donnés et des ressources externes avec lesquels l'environnement (capteurs et actionneurs, des objets, des objets intelligents etc.). Le niveau d'abstraction comble l'insuffisante conceptuel entre l'abstraction de l'agent et le niveau bas des détails du contexte de déploiement.
- Le service de l'environnement et la gestion des objets: Cette couche est responsable de la fonction principale des environnements et elle permet de leur gestion, un environnement définit la sorcière de services de base liés par ses ressources, l'environnement et sa découverte dynamique de l'objet.
- La composition du service de l'environnement et la composition de ses objets.

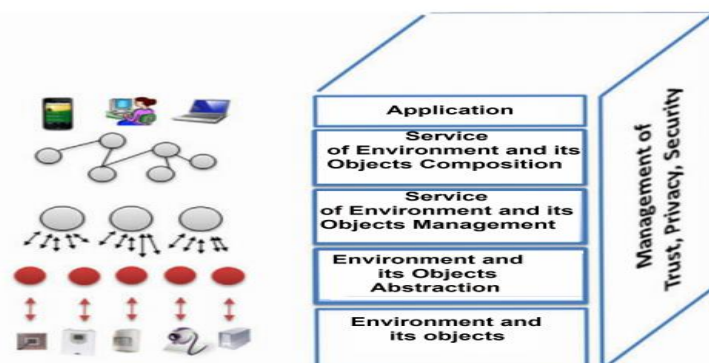


Figure 3.2 : Architecture IoT SOA proposé



Notre architecture

Environnement: l'environnement proposé dans notre système composé par le système de gestion de l'environnement, la composition des services de l'environnement, les services de l'environnement et de la communication de l'environnement.

- Gestion de système de l'environnement : le système de gestion de l'environnement gère un ensemble de procédures de fonction et les tâches dans l'environnement.

Environnement	Environment management
	Environment service composition
	Environment services
	Environment communication

Figure 3.3: couches Environnement

- Service de l'environnement: offre des services pour l'environnement
- La composition du service de l'environnement: Les services peuvent être composés afin d'exécuter une action.
- Communication de l'Environnement: Assurer l'inter et l'intra communication.

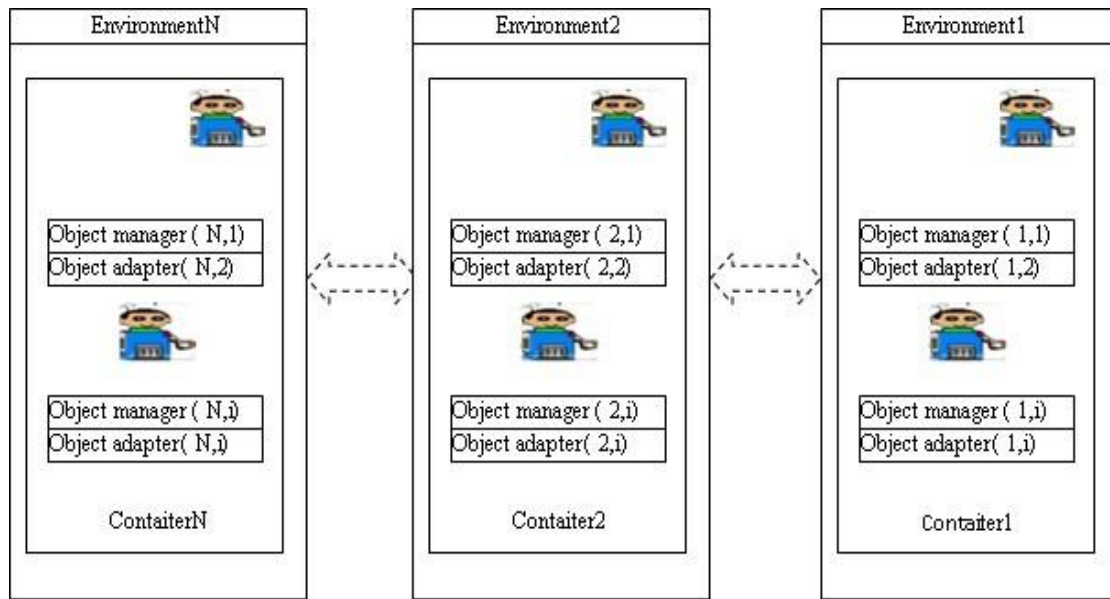


Figure 3.4 Architecture globale

Identificateur d'agent	Le nom d'un agent dans un Environment
Services d'agent	Liste des services d'agent
Object d'agent	l'Object en relation avec un agent.
Mobilité d'agent	code et activité responsable d'agent mobilisé.

Tableau 1: Relation Agent Objet et environnement

Agent Mobile

Le modèle d'agent mobile est proposé afin de faciliter la configuration du système dynamique, l'agent mobile est composé par identifiant de l'agent, des services d'agent, objet de l'agent, la mobilité des agents[43].

Un objet est représenté dans notre environnement par un adaptateur et gestionnaire d'objet et de l'objet services.



4. Diagramme d'Activité

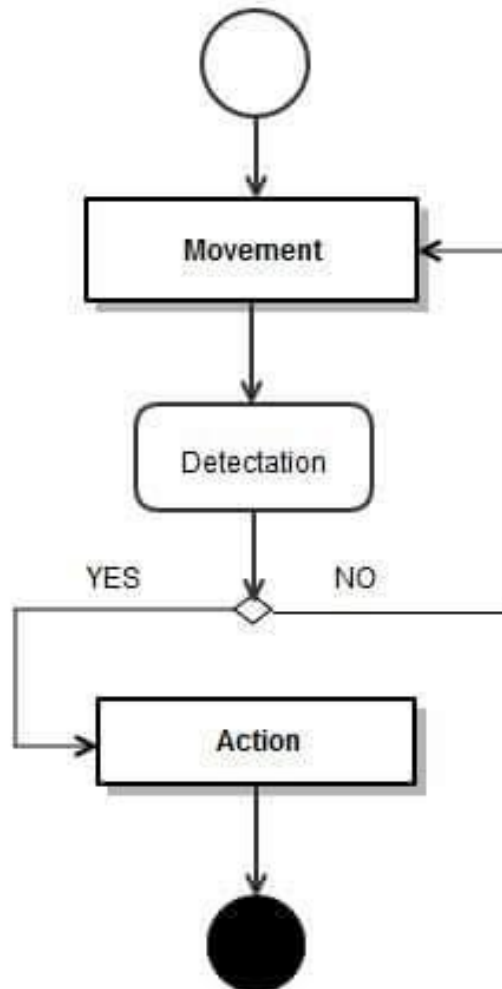


Figure 3. 5: Diagramme d'Activité

Si un agent détecte un événement (mouvement) l'agent réagi par l'action pertinente.



5. Diagramme de Séquence

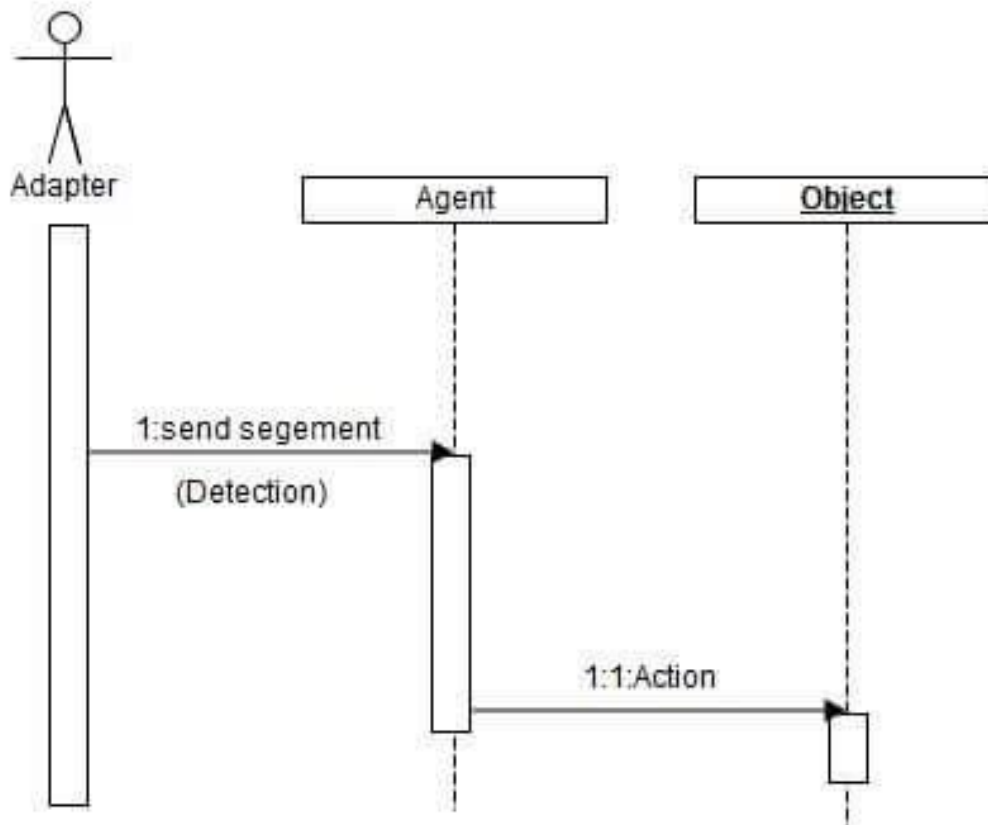


Figure 3.6 : Diagramme de Séquence

Si l'adaptateur détecte un événement, l'agent réagit par l'action pertinente



6. Diagramme de classe

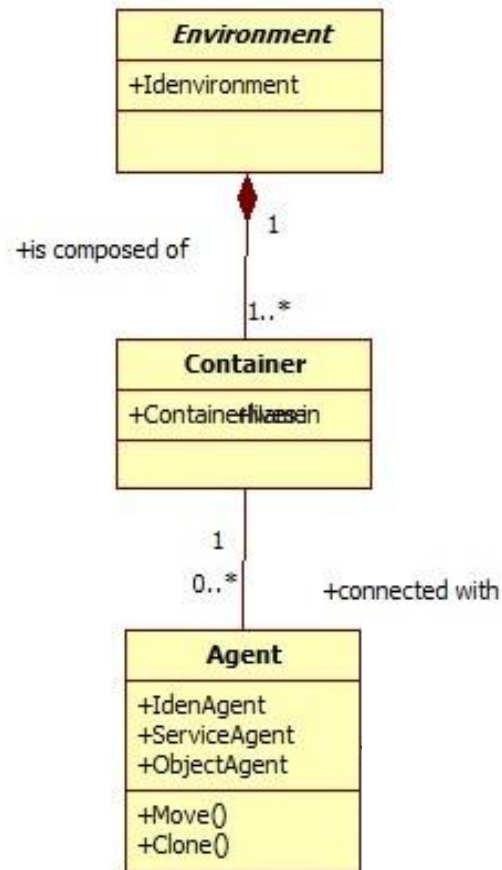


Figure 3. 7: Diagramme de classes

Conclusion

Dans ce chapitre nous avons présenté l'état de l'art de notre contribution et les travaux existants. Ensuite nous avons présenté l'architecture globale et enfin nous avons fait une modélisation de notre système par diagrammes d'Activité et de Séquence et diagramme de class.

CHAPITRE 04

Implémentation



Chapitre04: Implémentation

INTRODUCTION

Dans ce chapitre, nous allons aborder tout ce qui concerne le travail appliqué

Savoir Initialement langage de programmation utilisé et la langue de l'environnement de travail utilisé pour résoudre les formes et obtenir les résultats souhaités. Pour travailler sur le projet, nous avons utilisé Arduino.

1. QU'EST-CE QU'ARDUINO

Le système Arduino est un outil pour fabriquer de petits ordinateurs qui peuvent capter et contrôler davantage de choses du monde matériel que votre ordinateur de bureau. C'est une plateforme open-source d'électronique programmée qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel, véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.

Arduino peut être utilisé pour développer des objets interactifs, pouvant recevoir des entrées d'une grande variété d'interrupteurs ou de capteurs, et pouvant contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielles. Les projets Arduino peuvent être autonomes, ou bien ils peuvent communiquer avec des logiciels tournant sur votre ordinateur (tels que Flash, Processing ou Max MSP). Les cartes électroniques peuvent être fabriquées manuellement ou bien être achetées pré-assemblées; le logiciel de développement open-source peut être téléchargé gratuitement.

Le langage de programmation Arduino est une implémentation de Câblage, une plateforme de développement similaire, qui est basée sur l'environnement multimédia de programmation Processing.

Nous avons utilisé C++ est un langage de programmation compilé, permettant la programmation sous de multiples paradigmes comme la programmation procédurale, la programmation orientée objet et la programmation générique. Le langage C++ n'appartient à personne et par conséquent n'importe qui peut l'utiliser sans besoin d'une autorisation ou obligation de payer pour avoir le droit d'utilisation. C++ est l'un des langages de programmation les plus populaires, avec une grande variété de plates-formes matérielles et de systèmes d'exploitation.

2. POURQUOI ARDUINO

- **Pas cher** : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plateformes. La moins chère des versions du module Arduino peut être assemblée à la main, et même les cartes Arduino pré-assemblées coûtent moins de 25 Euros (microcontrôleur inclus...) !!!



Chapitre04: Implémentation

- **Multi-plateforme** : Le logiciel Arduino, écrit en Java, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.

- **Un environnement de programmation clair et simple**: L'environnement de programmation Arduino (= le logiciel Arduino) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puisse en tirer profit également. Pour les enseignants, il est basé sur l'environnement de programmation Processing : les étudiants qui apprennent à programmer dans cet environnement seront déjà familiarisés avec l'aspect du logiciel Arduino.

- **Logiciel Open Source et extensible** : Le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés. Le langage peut être aussi étendu à l'aide de bibliothèques C++, et les personnes qui veulent comprendre les détails techniques peuvent reconstruire le passage du langage Arduino au langage C pour microcontrôleur AVR sur lequel il est basé. De la même façon, vous pouvez ajouter du code du langage AVR-C directement dans vos programmes Arduino si vous voulez.

- **Matériel Open source et extensible** : Les cartes Arduino sont basées sur les microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, etc... Les schémas des modules sont publiés sous une licence Creative Commons, et les concepteurs de circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dans le but de comprendre comment elle fonctionne et pour économiser de l'argent.

3. Travaux pratiques :

Barrières de code spécial:

Elle est l'œuvre de ce dernier de se déplacer librement et lorsqu'ils sont exposés à aucune barrière de la manière et la dimension de code non autorisée dans la lumière d'alarme se produit par exemple, nous avons également affiché la distance que vous le garder hors de la barrière, et ainsi de suite ...

Chapitre04: Implémentation



```
Arduino IDE 1.6.7
مساعدة نوات الشيفرة البرمجية على هذا

Serbis
.....
Motor Shield I-Channel DC Motor Demo
by Randy Serafan

For more information see:
http://www.instructables.com/id/Arduino-Motor-Shield-Tutorial/
.....

#define trigPin 9
#define echoPin 8
#define ledPin 13

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;

  if (distance >= 50 || distance <= 0) {
    //Serial.println("Not in range");
    digitalWrite(ledPin, HIGH); // Light OFF
  }
}
```

Figure 4.1 interface de programme

Ces images montrent la machine et ceinturés

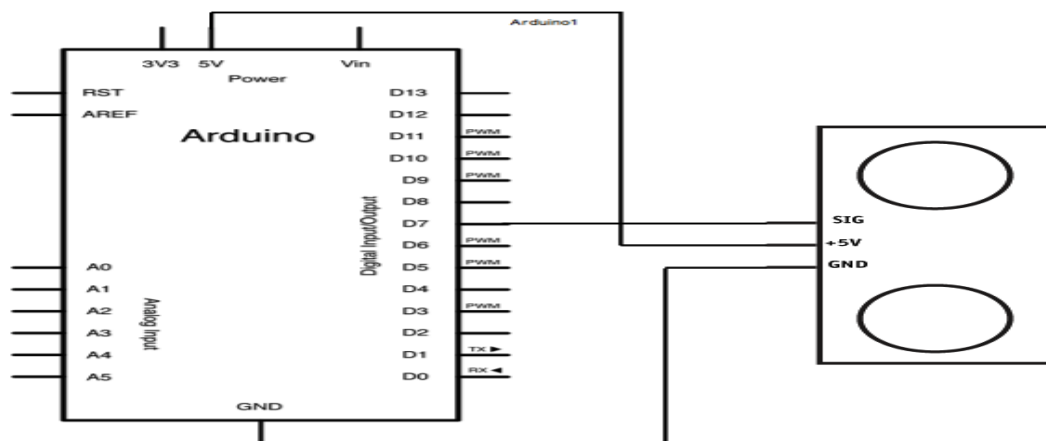


Figure 4.2 Circuit de notre programme de détection des obstacles.

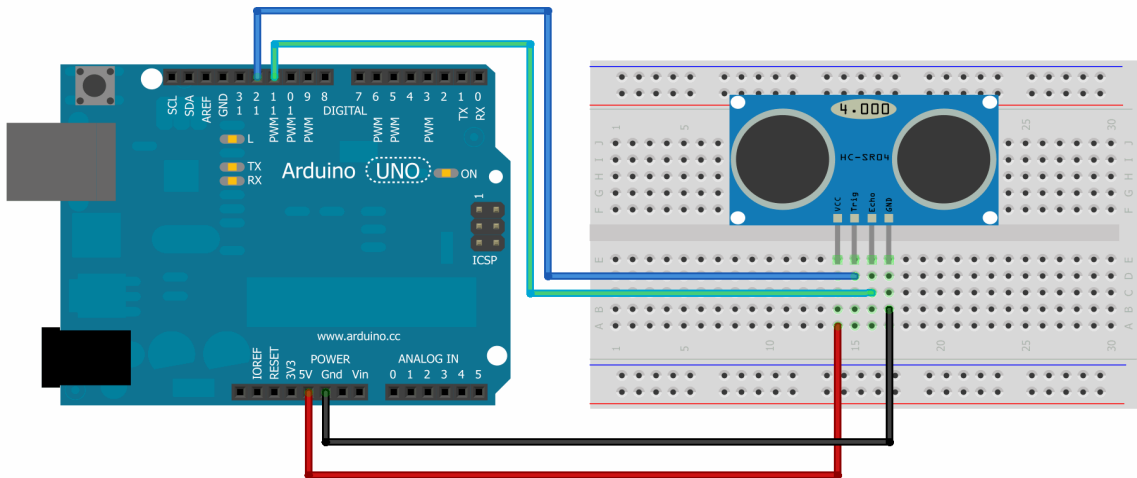


Figure 4.3 Carte Arduino et câblage de détection des obstacles.



Figure 4.4 Agent mobile (Panier)

L'agent mobile (panier) permet de capter les obstacles, afin de faciliter le mouvement de panier .

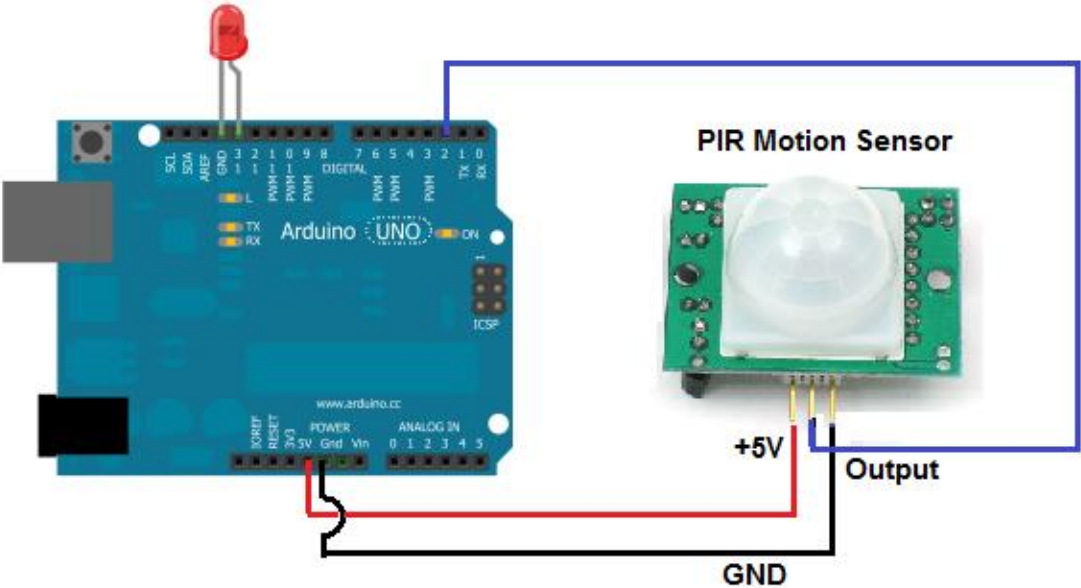


Figure 4.6 Carte Arduino et câblage de détection des mouvement.



Figure 4.7 Agent mobile détection des mouvement.

Chapitre04: Implémentation



Conclusion:

Dans ce chapitre, nous avons implémenté un système à base d'agent mobile, un agent joue le rôle d'un panier qui permet de capter les obstacles afin de faciliter le mouvement dans un centre de commerce. Le deuxième agent capter le mouvement d'entre et de sortie des paniers dans un centre de commerce.

Cette application envisage comme environnement celui du centre de commerce et celui de l'agent mobilisé de dans.

CONCLUSION GENERALE

CONCLUSION GENERALE

Notre travail propose une architecture pour L'internet des objet cette architecture prendre au niveau des modélisation la notion d'environnement.

L'environnement peut-être réalise des taches et des actions qui interagies avec les objet de l'environnement.

L'avantage de notre architecture permet de faciliter l'implémentation de système distribue.

Nous avons valide notre model par l'utilisation de carte Arduino dans un environnement de centre commercial.

Le perspective de notre travail et l'intégration de programme Arduino avec une plate forme a base d'agent.

BIBLIOGRAPHIE

Bibliographies

- [1] J. Dale. A mobile agent architecture to support distributed resource information management. Rapport technique, Department of Electronics and Computer Science.
- [2] M. Huhns et M. Singh. Agents on the web, mobile agents. IEEE Internet Internet Computing, pages 80-82, Mai-Juin 1997.
- [3] B. Venners. Solve real problems with aglets, a type of mobile agent. Rapport technique, Netscape World Magazine, Mai 1997.
- [4] I. Jarras, B. Chaïb-Draa Aperçu sur les systèmes Multi-Agents. Série Scientifique du centre inter universitaire de recherche en analyse des organisations -CIRANO-2002
- [5] H. Jens, G. Carmelita and F. Peyman. « Agent technology for the UMTS VHE concept ». ACM/IEEE MobiCom' 98, Workshop on Wireless Mobile Multimedia, Dallas, USA, October 1998
- [6] J. Vitek. Secure object spaces. In Proceedings of the 2nd ECOOP Workshop on Mobile Object Systems, pages 41-48, Linz, Austria, Juillet 8-9 1996
- [7] Farah Abdel Majid : « Vers un IDS Intelligent à base d'agents mobiles ». Mémoire de DEA. Pour obtenir le Diplôme des Etudes Approfondies en informatique Appliquée à la Gestion 2002/2003. Université de Tunis
- [8] Falout S.El : « Programmation répartie, optimisation par agent mobile », Thèse du doctorat soutenue le 29 Novembre 2006, de l'université de CAEN.
- [9] Christophe Cubat Dit Cros : « Agents Mobiles Coopérants pour les Environnements Dynamiques », Thèse de doctorat de l'Institut National Polytechnique de Toulouse, Soutenu le 2 Décembre 2005.
- [10] FIPA, <http://www.fipa.org>
- [11] Francis G. McCabe : OMG - FIPA Liaison, <http://www.objs.com/isig/omg-fipa-liaison-4.html>
- [12] Breugst M, Busse I et Magedanz T: « Grashopper – A Mobile Agent Platform for IN Based Service Environments ». Proceedings of IEEE IN Workshop, 1998
- [13] Amy L, Murphy, Gian Pietro Picco, Gruia-Catalin Roman et Lime: « A middleware for physical and logical mobility. In Proceeding of the 22nd International Conference on Distributed Computing Systems » (ICDCS 2001), Mesa, Arizona, USA, April 2001.
- [14] Danny B, Lange et Yariv Aridor: « Agent Transfer Protocol -ATP/0.1. IBM Corporation», 19 mars 1997
- [15] Dag Johansen, Kaare J.Lauvset, Robbert van Renesse, Fred B.Schneider, Nils P.Sudman et Kjetil Jacobsen: « A TACOMA retrospective Software & Practice and Experience », May 2002.
- [16] Akihiko Ohsuga, Yasuo Nagai, Yutaka Irie, Masanorie Hattori et Shinichi Honi-den: « PLANGENT an approach to making mobile agents intelligents. IEEE Internet Computing » July/August 1997.
- [17] <http://books.openedition.org/editionsms/84>

- [18] Christian Buckl, Stephan Sommer, Andreas Scholz, Alois Knoll, Alfons Kemper, Jörg Heuer, and Anton Schmitt, Services to the _eld : An approach fore source constrained sensor/actor networks, Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on, IEEE, 2009.
- [19] <http://jeanchavanat.wix.com/les-objets-connectes#!-propos1/c1uam>
- [20] FIPA, <http://www.fipa.org>
- [21] Mike P Papazoglou and Willem-Jan Van Den Heuvel, Service oriented Architectures: approaches, technologies and research issues, The VLDB journal 16(2007).
- [22] C. Peltz, Web services orchestration and choreography, Computer 2003.
- [23] David Booth, Hugo Haas, Francis McCabe, Eric Newcomer, Michael Champion, Chris Ferris, and David Orchard, Web services architecture, 2004.
- [24] Said HARCHI: «Un protocole de session dans les réseaux de capteurs sans fils», Thèse du doctorat soutenue le 06 décembre 2013, de l'université de lorraine.
- [25] <http://blog.nicolashachet.com/niveaux/confirmelarchitecture-rest-expliquee-en-5-regles/>
- [26] M. zur Muehlen, J.V. Nickerson, and K.D. Swenson, Developing web services choreography standards_the case of rest vs. soap, Decision Support Systems 40 (2005).
- [27] Nissanka B Priyantha, Aman Kansal, Michel Goraczko, and Feng Zhao, Tiny web services : design and implementation of interoperable and evolvable sensor networks, Proceedings of the 6th ACM conference on Embedded network sensor systems, ACM, 2008.
- [28] Yen-Kuang Chen, Challenges and opportunities of internet of things, Design Automation Conference (ASP-DAC), 2012 17th Asia and South Paci_c, IEEE, 2012.
- [29] http://www.memoireonline.com/05/12/5812/m_La-realisation-dune-application-de-contrle-total-des-processus-dun-ordinateur-distant8.html
- [30] Gerben G. Meyer, Kary Främling, and Jan Holmström, Intelligent products : A survey, Computers in Industry 60 (2009).
- [31] F. Curbera, Y. Golland, J. Klein, F. Leymann, S. Weerawarana, et al., Business process execution language for web services, version 1.1, (2003).
- [32] Luigi Atzori, Antonio Iera, and Giacomo Morabito, The internet of things : A survey, Computer Networks 54 2010.
- [33] <https://msdn.microsoft.com/en-us/library/dd170125.aspx>
- [34] Erdal Cayirci, Wireless sensor and actuator network applications and challenges, Autonomous Sensor Networks : Collective Sensing Strategies for Analytical Préposes 2013.
- [35] https://fr.wikipedia.org/wiki/Internet_des_objets#Environnements_intelligents
- [36] <http://coap.technology/>
- [37] <http://www.lemag-numerique.com/2016/04/iot-systemes-dexploitation-fleurissent-8682>

- [38] Niels Brouwers, Koen Langendoen, and Peter Corke, Darjeeling, a featurerich for the resource poor, Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, ACM, 2009.
- [39] Hydra Middleware Project, FP6 European Project, <http://www.hydramiddleware.eu>
- [40] A. Barros, M. Dumas, and P. Oaks, Standards for web service choreography and orchestration : Status and perspectives, Business Process Management Workshops, Springer, 2006.
- [41] D. Giusto, A. Iera, G. Morabito, L. Atzori (Eds.), The Internet of Things, Springer, 2010. ISBN: 978-1-4419-1673-0. Fortino, G., Guerrieri, A., Russo, W.: Agent- oriented smart objects development. In: 16th IEEE International Conference on Computer Supported Cooperative Work in Design, pp. 907-912 (2012).
- [42] Kortuem, G., Kawsar, F., Fitton, D., Sundramoorthy, V.: Smart objects as building blocks for the internet of things. Internet Comput. 14 (1), 4451 (2010).
- [43] Leppnen, Teemu, et al. : Mobile agents-based smart objects for the internet of things. Internet of Things Based on Smart Objects. Springer International Publishing, 2014. 29-48.
- [44] Shelby, Z. : Embedded web services. IEEE Wireless Commun. Mag. 17 (6), 5257 (2010). [8] Mars-Maestre, Iván, et al. : Mobile agents for service personalization in smart environments. Journal of Networks 3.5 (2008): 30-41.