INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

DOCTORALE SCHOOL: *IAEM*

Loria laboratory

THESIS

Presented at 2011/07/06 in partial fulfillment of the requirements
for the degree of Docteur de l'INPL

(Speciality : Computer science)

by

Shahram NOURIZADEH

A context-aware and QoS-aware telehomecare system

Supervisors:	Ye-Qiong Song	Professor (INPL - Nancy)
	Jean-Pierre Thomesse	Professor (INPL - Nancy)

<u>Defense Committee</u>:

Chairman:	Norbert Noury	Professor (INSA - Lyon)
Reviewers:	Norbert Noury	Professor (INSA - Lyon)
	Bessam Abdulrazak	Adjoint Professor (University of Sherbrooke QC)
Examinators:	David Daney	Chargé de recherche (INRIA Sophie-Antipolis)
	Ye-Qiong Song	Professor (INPL - Nancy)
	Jean-Pierre Thomesse	Professor (INPL - Nancy)
Invited member:	Claude Deroussent	Doctor of Medicine (MEDeTIC - Colmar)

NSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

ÉCOLE DOCTORALE : *IAEM*

Laboratoire de *Loria*

THÈSE

Présentée et soutenue publiquement le 06/07/2011

pour l'obtention du grade de Docteur de l'INPL

(Spécialité : Informatique)

par

Shahram NOURIZADEH

Un système de télésanté contextuel avec support de qualité de service pour le maintien à domicile

Directeur de thèse :	Ye-Qiong Song	Professeur (INPL - Nancy)
Co-directeur de thèse :	Jean-Pierre Thomesse	Professeur (INPL - Nancy)

Composition du jury :

Président du jury :	Norbert Noury	Professeur (INSA - Lyon)
Rapporteurs :	Norbert Noury	Professeur (INSA - Lyon)
	Bessam Abdulrazak	Professeur adjoint (Université de Sherbrooke)
Examinateurs :	David Daney	Chargé de recherche (INRIA - Sophie-Antipolis)
	Ye-Qiong Song	Professeur (INPL - Nancy)
	Jean-Pierre Thomesse	Professeur (INPL - Nancy)
Membres invités :	Claude Deroussent	Docteur en médecine (MEDeTIC - Colmar)

Table of contents

List of Figures

List of Tables

Index

Acknowledgments

This thesis arose in part out of years of research that has been done since I came to trio's group. By that time, I have worked with a great number of people who contributed the research in assorted ways and merited the thesis special attention. It is a pleasure to convey my gratitude to them my acknowledgment with all my respect.

First of all, I would like to announce my gratitude to Pr. Y.Q. Song for his supervision, advice, and guidance from the very early stage of this research. Above all and the most needed, he encouraged me unsparingly and supported me in various ways. His truly scientific intuition has made him as a constant paragon of ideas and passions in science, which exceptionally inspired me and enriched my growth as a student, a researcher and a scientist want to be.

I gratefully acknowledge Pr. J.P. Thomessse for his advice, supervision, and crucial contribution, which made him a backbone of this research and the thesis as well.

I gratefully thank Pr. Noury and Pr. Abdulrazak for taking their precious time to read this thesis and having their constructive comments on this thesis. I am thankful that in the midst of all their activity, they accepted to be members of the reading committee.

His original involvement has triggered and nourished my intellectual maturity that I benefited for a long run.

Many thanks go in particular to Pr. Mme. Simonot for her valuable advice in science discussion.

My special thanks go to Pr. Mery and M. Quinson for their valuable helps during my education at Nancy.

Abstract

This thesis is a thesis CIFRE between LORIA and the MEDETIC Company and focuses on the design of telehomecare system for the elderly.

In addition to the design of a remote surveillance architecture "Maisons Vill'Âge" based on networks of heterogeneous sensor (home automation, IEEE802.15.4/Zigbee, Wifi, Bluetooth), thesis has contributed essentially on the proposal of a clustering and routing protocol in the network of wireless sensors with an approach to fuzzy logic, and of a middleware for the collection and processing of data from sensors with the management of the quality of service as a special feature.

A first platform was developed at Colmar (MEDETIC) and a second who is more complete is under development at LORIA (http://infositu.loria.fr/).

By using this system, MEDeTIC, offers a new concept of smart homes for the senior citizens, named in French "Maisons Vill'Âge". The first housing schemes are being built in 2 departments of France. A flat is entirely equipped to act as a demonstrator and as laboratory of research and development.

The system is designed for the elderly who wish to spend their old age in their own home, because of its potential to increase independence and quality of life. This would not only benefit the elderly who want to live in their own home, but also the national health care system by cutting costs significantly.

Based on this PhD thesis, MPIGate, a "Multiprotocol Interface and Gateway for for telecare, environment", has been developed. MPIGate was awarded in the competition of the Ministry of Higher Education and Research and OSEO 2010.

Keywords:

Telehomecare, Activity monitoring, Context-aware Middleware, Wireless Sensor Networks, Quality of Service (QoS), Adaptive QoS, Adaptive Routing, Clustering, Home automation, Smart home, Fuzzy Logic

Résumé

Cette thèse est une thèse CIFRE entre le LORIA et la société MEDETIC et porte sur la conception des systèmes de télésurveillance pour le maintien à domicile des personnes âgées.

Le système est conçu aux personnes âgées qui veulent passer leur vieillesse dans leur propre maison, à cause de son potentiel pour augmenter l'indépendance et la qualité de la vie. Cela profiterait non seulement aux personnes âgées qui veulent vivre dans leur propre maison, mais aussi le système de santé publique en coupant des prix de façon significative.

Mis à part la conception d'une architecture de télésurveillance « Maisons Vill'Âge » basée sur des réseaux de capteurs hétérogènes (Domotique, IEEE802.15.4/Zigbee, Wifi, Bluetooth), la thèse a contribué essentiellement sur la proposition d'un protocole de clustering et de routage dans le réseau de capteurs sans fil avec une approche de la logique floue, et d'un middleware pour la collecte et le traitement des données des capteurs avec la gestion de la qualité de service comme particularité.

Une première plateforme de test a été développée à Colmar (MEDETIC) et une seconde, plus complète et fait suite de cette thèse, est en cours de développement au LORIA (http://infositu.loria.fr/).

Nous avons participé dans le concours de ministère de l'Enseignement supérieur et de la recherche avec un projet intitulé MPIGate: « Multi Protocol Interface and Gateway for telecare, environment Monitoring and Control » et nous avons gagné le prix de ce concours au niveau d'émergence.

Mots clés :

Télésanté, Actimétrie, Intergiciel contextuel, Réseaux de capteurs sans fils, Qualité de Service (QdS), QdS adaptative, Routage adaptative, Clustering, Domotique, Maison intelligente, Logique floue

Résumé étendu en Français

I. Contexte de travail

Comme la population vieillit, les gens deviennent de plus en plus dépendants leurs capacités physiologiques, leurs fonctions sensorielles, motrices et cognitives se détériorent; ces changements liés à l'âge, sont amplifiés si ils sont accompagnés de conditions pathologiques courantes chez les personnes âgées.

L'accroissement du nombre de personnes âgées de plus de 75 ans constitue un enjeu majeur pour les politiques publiques. C'est en général autour de cet âge que débute la perte d'autonomie, c'est-à-dire la difficulté accrue à choisir librement son mode de vie.

Les personnes âgées commencent alors à avoir besoin d'une aide, à laquelle la solidarité familiale ne parvient pas toujours à répondre. Alors que la part des 60-74 ans s'est stabilisée depuis vingt ans, les personnes de plus de 75 ans sont aujourd'hui un tiers de plus, soit 5 millions, et le nombre des plus de 85 ans a doublé, pour atteindre 1,3 million de personnes.

La France devrait connaître un vieillissement un peu moins marqué que ses voisins européens. Dès 2011 cependant, la part des personnes de plus de 60 ans dépassera celle des moins de 20 ans. D'ici 2030, le nombre de personnes de plus de 60 ans passera de 13 à 20 millions.

Elles représenteront alors près d'un Français sur trois. Si ces tendances se poursuivent jusqu'en 2050, le nombre de personnes de plus de 60 ans va doubler, celui des personnes de plus de 75 ans tripler, tandis que les personnes de plus de 85 ans verraient leur nombre quintupler. Les centenaires, quant à eux, passeraient de 9.000 à plus de 100.000.

Les villes devront également faire face à un vieillissement marqué. Selon l'INSEE, les aires urbaines de plus de 50.000 habitants connaîtront un vieillissement plus marqué que le reste de la France. Alors que le nombre de personnes âgées progresserait de 18% à la campagne d'ici 2015, il augmenterait de 35% dans les aires urbaines moyennes (50.000 à 900.000 habitants) et de 30% dans les grandes métropoles.

De nos jours, beaucoup de personnes âgées dépendantes doivent quitter leur maison et déménager dans des centres de soins ou d'hébergements spéciaux. Ce qui n'est pas souvent bien accepté par eux en raison du coût élevé et l'isolement vis-à-vis de leur famille et de leur environnement habituel.

Les nouvelles technologies d'information et communication peuvent jouer un rôle clé dans le développement d'un habitat intelligent pour permettre aux seniors de bénéficier d'un environnement sécuritaire à leur domicile avec diverses activités et l'accompagnement des services pour leurs démarches quotidiennes, retardant ainsi au maximum le départ en maison de retraite ou centre de soins.

Pour ces raisons, de nombreux systèmes ont été développés depuis plusieurs années : les télésoins à domicile, la vidéoconférence, les systèmes de télé-consultation, des systèmes de détection d'alertes,… mais la plupart d'entre eux ne fournissent que des solutions partielles pour les cas spécifiques.

Donc, nous avons cherché à proposer un système de télésoins à domicile générique, modulaire et évolutif incluant un système d'actimétrie, considérant d'importants défis techniques et scientifiques tels que :

- la QoS,
- l'interopérabilité

- la facilité d'accès aux données pour développer des applications intelligentes pour la télésanté qui sont les principaux défis à relever dans ce type de systèmes.

II. Motivation

Les systèmes actuels sont limités en termes de perspectives commerciales et opérationnelles et souffrent de certains manques:

- Les projets et des produits n’ont pas encore donné naissance à une solution efficace qui permettrait de répondre aux différents besoins et attentes des utilisateurs.
- Les coûts de développement R&D nécessaires pour une adaptation propre à chaque communauté d'utilisateurs, sont très élevés.
- Les délais de mise sur le marché sont incertains et aucun modèle d'affaires n’a encore été évalué avec succès dans ce domaine.
- Les progrès des systèmes, aux niveaux institutionnel, éthique, juridique, financier et organisationnel restent insuffisants, et empêchent le développement des services de surveillance à domicile.
- La population n'est pas assez confiante, ni familiarisée avec ces solutions fondées sur l'innovation, bien que leur développement exige qu'ils deviennent une composante naturelle de l'environnement médical et social.

III. Défis

Dans un système de télésoins à domicile et de télésurveillance de l'activité, plusieurs défis doivent être abordés. Les défis technologiques qui concernent les caractéristiques générales des réseaux de capteurs sans fil (par exemple la mobilité, l'échec de capteur ou le lien de communication, des ressources énergétiques limitées ...), interopérabilité pour permettre une interaction souple entre différentes technologies hétérogènes installées dans le système et assurer les besoins exprimés en qualité de service.

À cet égard, la méthodologie de l'approche, le développement et l'intégration des technologies doivent tenir compte de certains concepts importants, tels que: la sensibilité au contexte et l'auto-adaptation, la nécessité d'une qualité de service (QdS) dynamique, les capacités d'évolution et de personnalisation. Dans cette thèse, nous nous concentrons uniquement sur la QdS, la sensibilité au contexte et l'auto-adaptation.

A. Non-interopérabilité entre les technologies

Les réseaux de capteurs hétérogènes sont devenus populaires. Dans les réseaux de capteurs hétérogènes, généralement, la détection est effectuée avec un grand nombre de nœuds peu coûteux, tandis que quelques nœuds suffisent pour filtrer les données, gérer leur fusion et le transport.

Ce cloisonnement des tâches assure un rapport coût-efficacité ainsi qu'une mise en œuvre plus efficace de la demande globale de mesure et de détection d'événement. Un exemple réel d'utilisation d'un réseau de capteurs hétérogènes est le contrôle des actionneurs domotiques dans les maisons intelligentes.

B. Quelité de Service (QdS)

La notion de qualité de service dans un réseau est liée à la différenciation de service (souvent par la définition de priorité). Le problème est qu'une même donnée n'a pas forcément la même importance selon son utilisation et son contexte. En plus la disponibilité de la ressource de communication sans fil est typiquement fluctuante dans le temps à cause des perturbations de l'environnement et les périodes de veille des nœuds (économie de l'énergie).

Il est donc important de disposer d'un algorithme qui prend en compte les besoins de l'application et alloue dynamiquement les ressources appropriées (priorité des paquets de données, choix de capteurs et de chemins du routage, …), et ceci en considérant la disponibilité des ressources réseaux en temps réel.

L'optimisation de la consommation d'énergie est l'un des défis de qualité de service dans les capteurs sans fil. La technologie sans fil la plus utilisée dans les réseaux à domicile est WiFi. Mais Wi-Fi consomme trop d'énergie. En raison de ce problème, il y a avantage à utiliser des technologies à faible consommation comme ZigBee, pour lesquels il est de nombreux mécanismes d'optimisation de consommation d'énergie.

C. Sensibilité au contexte et d'auto-adaptation

Les systèmes de santé représentent une classe d'applications capables de s'adapter dynamiquement à des changements d'environnement et des conditions d'utilisation ainsi qu'à l'indisponibilité d'un des dispositifs ou des services. Les systèmes sensibles au contexte et auto-adaptatifs sont des solutions intéressantes.

IV. Contributions de la thèse

Cette section décrit brièvement les principales contributions de ce travail de thèse. Les publications résultant de cette thèse sont énumérées dans les références [1-10].

A. «Maisons Vill'Âge ®": Un système de télésoins à domicile pour les personnes âgées [1-5]

Dans cette thèse, nous avons développé, "Maisons Vill'Âge ®», un nouveau concept de construction de maisons intelligentes grâce à l'intégration de la télémédecine et des systèmes de domotique. Le segment de population ciblé est constitué des personnes âgées et à mobilité réduite.

Nous avons conçu et développé une architecture distribuée pour ce système adapté aux personnes âgées qui souhaitent passer leur retraite dans leur propre maison, afin de préserver leur indépendance et leur qualité de vie. De tels systèmes devraient être aussi profitables au système national de santé en réduisant les coûts de manière significative.

Les premières maisons sont en construction, par l'association MEDeTIC, intégrant des outils de base de l'acquisition de données et des interfaces homme-machine. Un appartement est entièrement équipé en Alsace à titre de démonstration et de laboratoire.

La figure *a*, montre l'architecture générale du système, composée de différentes parties : le Réseau domotique, le Réseau de capteurs médicaux, le Réseau de caméras sans fils (voir partie 5. Conclusion), la Base de données médicales et les différentes plateformes du système: Plateforme serveurs - Centre d'appel - Plateforme de contrôle et d'accès à distance.

Ce système utilise des capteurs domotiques et d'autres capteurs d'environnement comme des détecteurs de présence au lit, sur la chaise, afin de contrôler le niveau d'activité des personnes âgées (Actimétrie). Les données d'activité sont analysées ensuite par une application intelligente qui est fondée sur la Logique Floue pour trouver des activités inhabituelles.

Le système est équipé de détecteurs médicaux sans fil pour contrôler l'état de santé des personnes âgées; il utilise également un réseau de détecteurs sans fil pour détecter les chutes. Le système permet de détecter des anomalies de santé en fixant des seuils d'alertes des données physiologiques mesurées.

D'autre part, l'interface de ce système propose un certain nombre de fonctionnalités comme : La Visiophonie, la Télévision, la Vidéo et la Musique à la demande, l'Internet et enfin le Contrôle domotique permettant à la personne âgée d'activer simplement les fonctionnalités proposées.

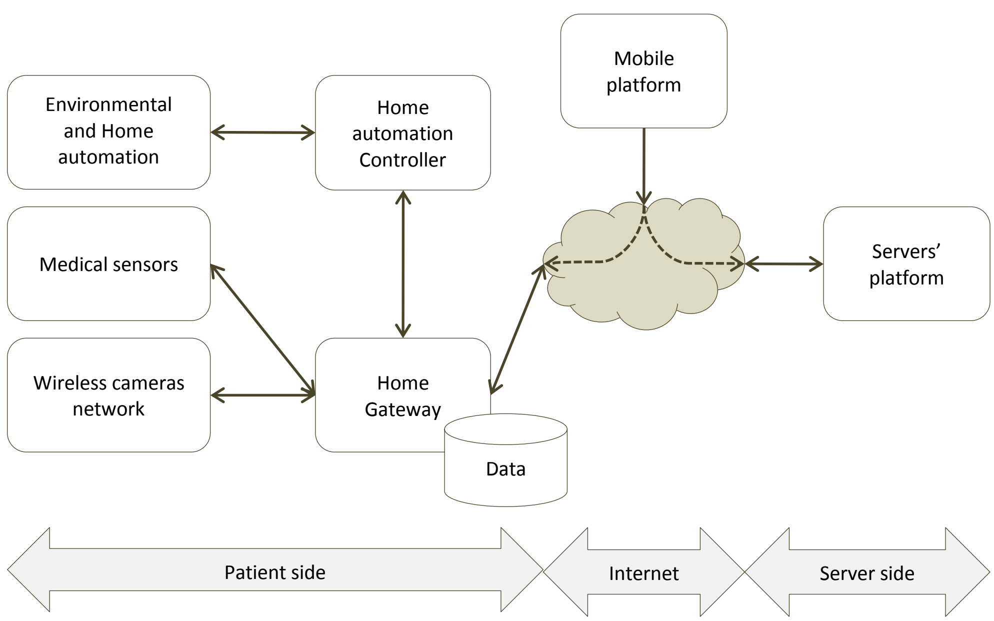


Figure a. Architecture générale du système

i. La simulation

La simulation consiste à reproduire le comportement et le fonctionnement de système dans un environnement informatique. Pour cela, nous avons développé un simulateur non piloté qui se déploie sur un ordinateur, sous Microsoft Windows sur la technologie de Visual C#.Net. Les paramètres et les conditions appliquées dans cette simulation sont :

- Un appartement de type F2 avec une surface de 34 m²: 1 chambre de 10 m², SdB de 4 m² et WC 3 m², une Salle de 12 m² (salon/salle à manger), Cuisine 5 m²;
- 1 détecteur de présence dans chaque pièce ;
- 1 détecteur de présence dans le lit (dans la chambre) ;
- 1 détecteur de présence dans le fauteuil (dans la salle) ;
- 1 personne ;
- Les règles statiques, sans un algorithme intelligent ;
- La vitesse moyenne de déplacement de la personne: 0,5 mètre/seconde ;
- Anomalie (Chute) pattern : « Random pattern » ;
- « Simulation round » : 100 rounds et 1 anomalie aléatoire dans chaque round. 1 Round est équivalant de 24 h.

a. Les différents types de configuration matérielle de la simulation

- **Type1** : Simulation avec seulement les capteurs de présence (1 capteur par pièce) ;
- **Type 2** : Simulation avec les mêmes capteurs de présence et 1 capteur de présence dans le lit ;

- **Type 3** : Simulation avec les mêmes capteurs de présence, 1 capteur de présence dans le lit et 1 capteur de présence dans le fauteuil du salon.

Le but est de trouver le pourcentage de fausses alarmes et d'anomalies détectées, dans les différentes configurations.

Tableau 1. SIMULATION RESULTS

Type	**Les situations anormales réelles**	**Les alertes déclenchées par le système**	**Fausse alertes**	Alertes réelles
1	100	180	149	31
2	100	135	88	47
3	100	114	50	64

b. Les résultats et leur analyse

La table 1 montre les résultats des simulations dans les 3 types de configuration. Même si, en fait, avec les simulateurs, on est dans un environnement idéal (sans échecs des capteurs, pas de perte des paquets d'informations), il y a une corrélation forte entre les résultats de simulation et la réalité, ce qui montre que les adaptations faites dans l'algorithme proposé initialement permettent d'améliorer très sensiblement les résultats.

Les résultats en simulateur montrent donc que, pour avoir un système capable d'avoir un maximum de détection des anomalies réelles, avec le moins de fausses alarmes, il faut :

- Ajouter les capteurs de présence dans le lit et le fauteuil, car ils augmentent le nombre de chutes détectées et réduiront le nombre de fausse alarmes ;

- Avoir une programmation et un algorithme intelligents, comme par exemple l'utilisation conjointe de logique flou;
- Passer par une phase d'apprentissage des habitudes du résident.

ii. Discussion

En comparaison aux systèmes traditionnels de soins médicaux, les systèmes de télémédecine restent des systèmes complexes, parce que ces différents systèmes doivent être interfacés les uns aux autres, fonctionnant parfois selon un type d'OS (operating system) différent et présentant des protocoles d'échanges (M2M) très sensiblement distincts selon leur nature (capteurs sans fil – protocoles ZigBee, Bluetooth et WiFi, capteurs domotiques - protocoles KNX ou IHC, …).

La méthodologie d'approche et le développement doivent intégrer toutes ces dimensions que sont: l'Interopérabilité, la Maintenabilité, l'Extensibilité et l'évolution des systèmes. D'autre part, dans le projet, on s'est attaché à développer une interface ergonome et simple d'utilisation, garantissant un niveau maximum d'appropriation de la part des utilisateurs.

L'interface, son accessibilité et son ergonomie ont été respectées au plus près des attentes et remarques des utilisateurs analysées en amont du développement de l'interface et ont démontré le grand degré d'importance dans ce type de développement. Figure *b* montre la partie santé de l'interface de notre système.

Les premières études montrent que tous les professionnels de santé ont présenté une attitude positive vers l'utilisation de l'application de TIC (Technologies de l'information et de la communication).

La majorité des personnes âgées qui ont été interrogées, déclarent un vif intérêt à l'utilisation de ces nouvelles technologies de TIC, seul 16 % d'entre elles restent défavorables à son utilisation. Notre étude montre également que la connaissance limitée de TIC ne présente pas un point favorable ou défavorable pour son utilisation.

Figure b. Interface utilisateur – Partie Santé

B. Support de la QoS dans les réseaux de capteurs sans fil [6-9]

La qualité de service (QdS) dans un réseau de capteurs et actionneurs hétérogènes pose un autre problème important à résoudre. En effet, la nature des flux transporté par un tel réseau est diverse. On y retrouve de la vidéo, des données de présence et de mouvement, des données liées à l'état de santé, des alertes sur des situations critiques (e.g. détection d'une chute), etc.

Cette variété de flux temps réel et non temps réel, nécessite une politique de qualité de service gouvernant le transport de ces différents flux selon leur priorité. Cette politique est d'autant plus indispensable en situation de crise où un délai de réaction excessif peut engendrer des conséquences intolérables. L'état de l'art actuel ne donne pas de solution satisfaisante lorsqu'il s'agit de communication sans fil et encore moins lorsqu'il s'agit d'un réseau hétérogène.

Dans cette thèse, un algorithme de routage basé sur la logique floue est proposé. Une récupération de route stable, un haut taux de livraison des données et l'équilibrage de charge entre les nœuds sont les principales caractéristiques de cet algorithme. L'algorithme utilise un système de décision multicritère pour former les clusters, gérer la mobilité et la disponibilité des nœuds.

Ce protocole s'adapte également à la mobilité et à l'indisponibilité des nœuds. Il est particulièrement efficace dans les réseaux qui utilisent des nœuds de capteurs pour l'agrégation de données, et dans lesquels le taux de livraison des données est important, comme dans le cas des réseaux de capteur médicaux pour la télésurveillance de l'état de santé.. Les résultats des simulations montrent que le protocole proposé est bien adapté pour de telles applications.

i. Motivation

Les réseaux de capteurs sans fil ont de nombreuses applications. Les maisons intelligentes, des mondes virtuels interactifs ne sont que quelques exemples. Un exemple est le réseau de santé.. Dans un tel réseau, en raison de la mobilité des patients, tous les capteurs qui leur sont attachés, sont mobiles ; donc la gestion de la mobilité et la localisation sont des points importants dans ces réseaux. La figure *c*, montre une architecture typique d'un système de soins. Dans ce système, l'épine dorsale de communication comprend les nœuds mobiles ou fixes et aussi les nœuds fixes ou filaires ou capteurs attachés aux autres personnes, comme les médecins et les infirmières.

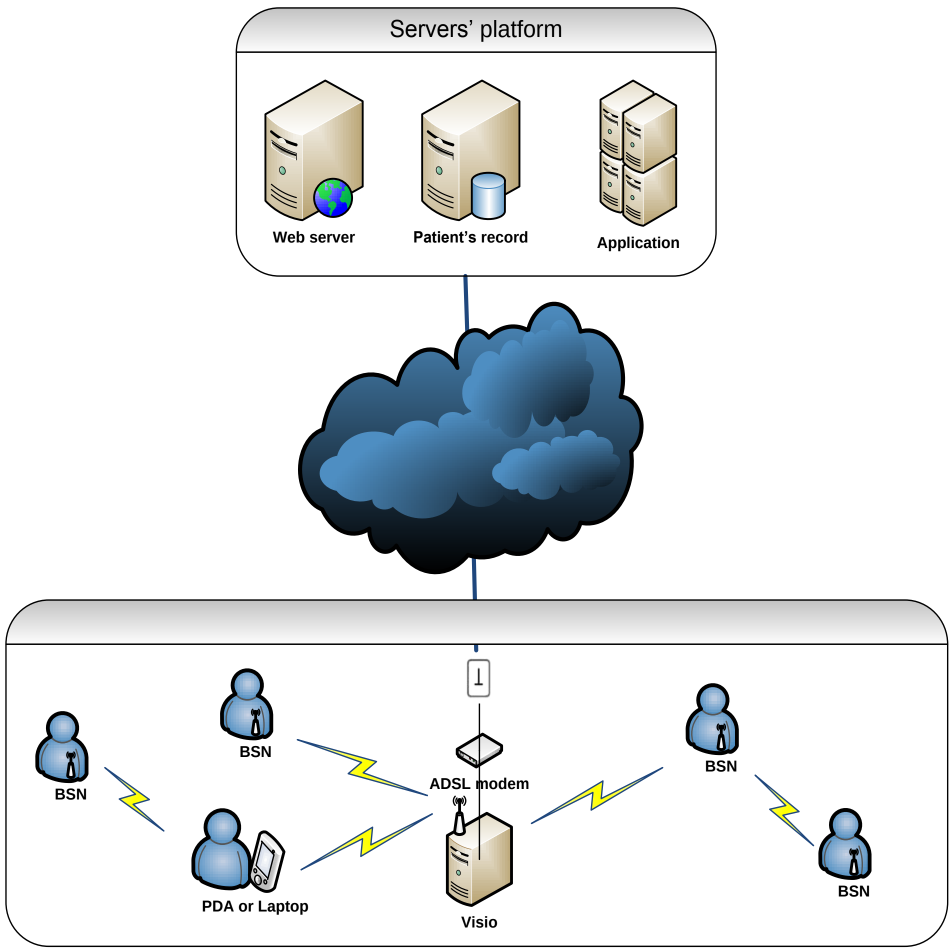


Figure c. An example of mobile network

En fait "Maisons Vill'Âge ®" est un système plus complexe. Le premier objectif est de fournir des solutions de QoS pour ce concept. La figure *d* présente l'architecture du système général de "Maisons Vill'Âge ®".

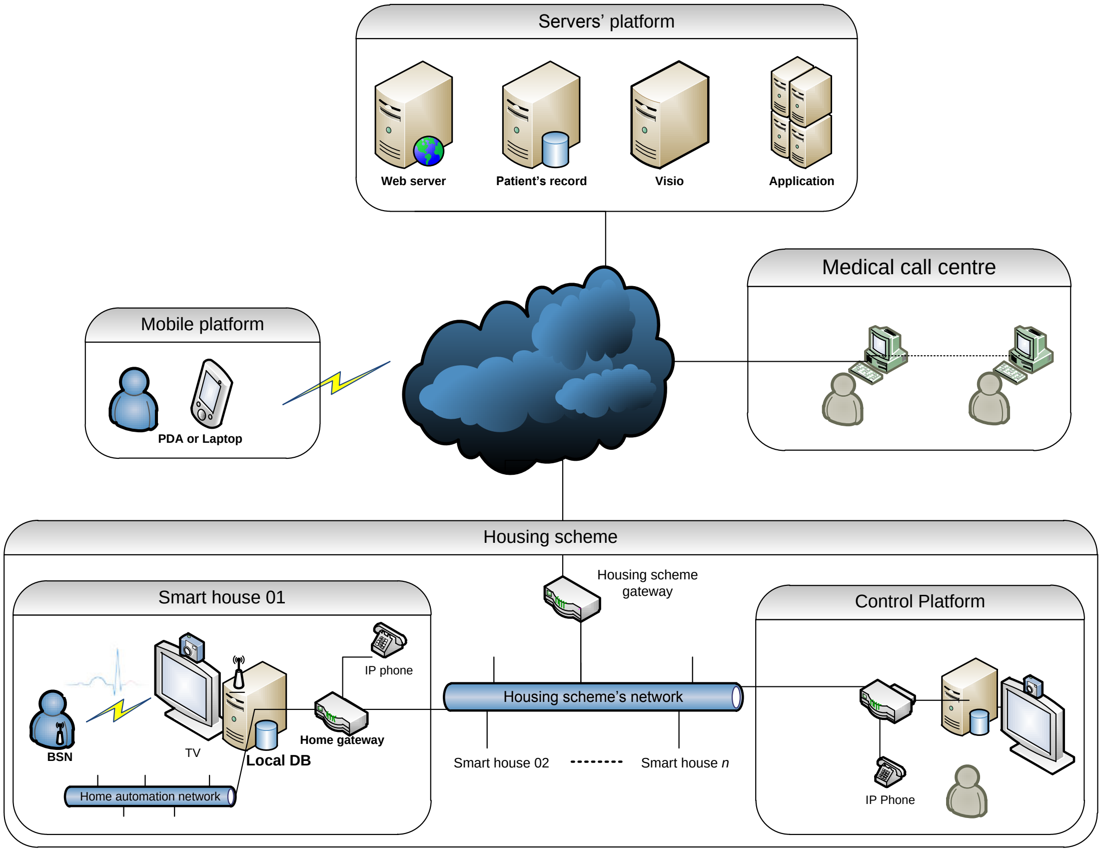


Figure d. System architecture - "Maisons Vill'Âge®"

Ces systèmes ont besoin d'un routage adaptatif et d'une localisation pour soutenir la mobilité des capteurs et des changements de topologie de réseau lorsque les patients se déplacent.

ii. Les défis et les objectifs

En raison des avantages des réseaux hiérarchiques ou basés sur le concept de cluster liés à l'évolutivité et à l'efficacité de la communication, nous avons choisi ce type d'architecture de réseau pour "Maisons Vill'Âge ®" (figure *e*).

Comme le montre la figure *e*, les utilisateurs et les capteurs médicaux sont mobiles et les capteurs environnementaux fixes peuvent être avec ou sans fil. Quand l'utilisateur quitte son appartement, le système doit garantir la localisation continue et la surveillance de la santé de la personne.

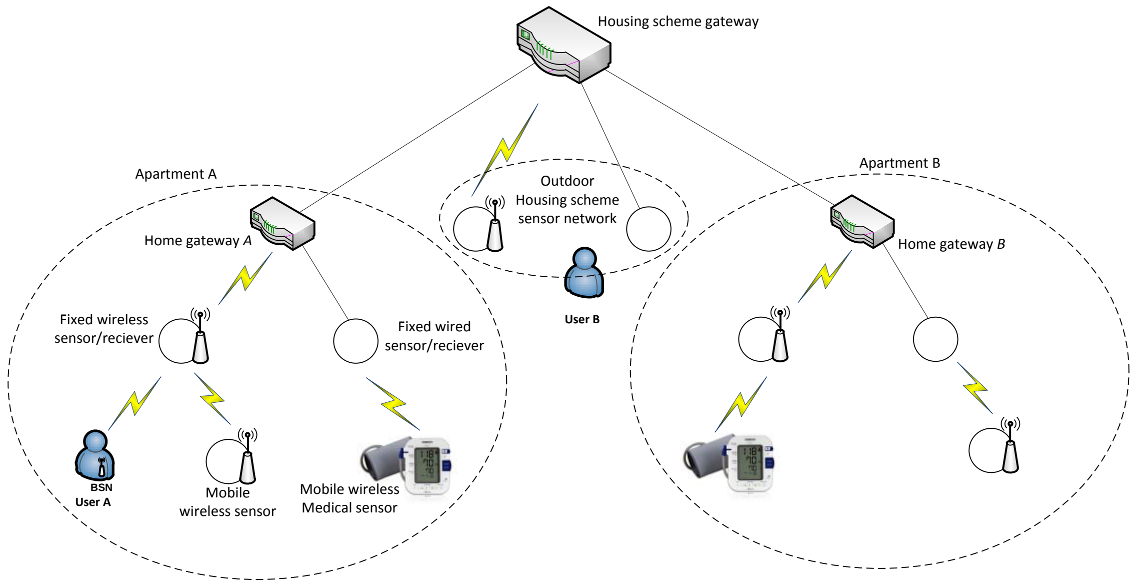


Figure e. System architecture - “Maisons Vill’Âge®”

À cette fin, il y a un réseau de capteurs extérieurs qui comprend des nœuds de capteurs avec ou sans fil. Il y a un écart considérable entre les technologies de réseau de capteurs actuelles et les besoins particuliers des systèmes de télésoins à domicile. Généralement, le système doit être capable de fournir des exigences QoS en considérant les défis techniques de WSNs, afin de fournir un regroupement dynamique robuste et routage (voir figure *f*).

À cet égard et en ce qui concerne nos objectifs, les exigences QdS doivent être considérées. Les exigences QdS diffèrent d'une application à une autre. Dans le cadre de cette thèse, elles peuvent être spécifiées en termes d'optimisation de la consommation énergétique, d’équilibrage de charge, ou en minimisant le retard.

Nos recherches nous ont conduits à proposer FLoR, un nouvel algorithme de routage qui peut gérer la mobilité et l'indisponibilité (causée par l'échec ou la mobilité) des nœuds. L'algorithme utilise un mécanisme de décision multicritère pour former les clusters et gérer la mobilité et l'indisponibilité des nœuds. FLoR est totalement distribué.

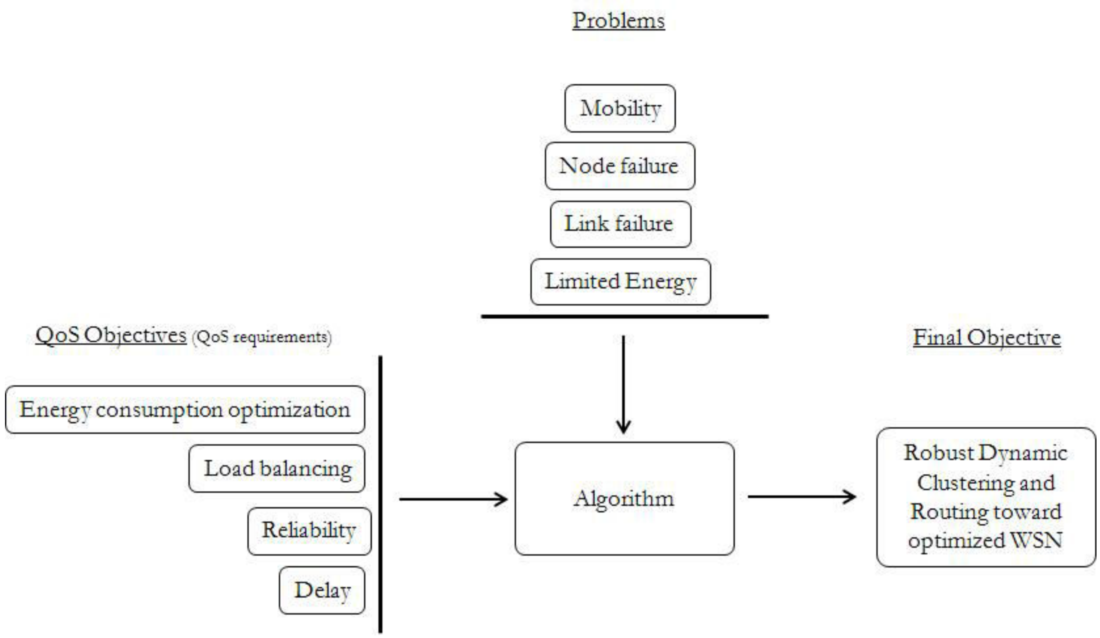


Figure f. Problem description

iii. FLoR – Fuzzy Logic based Adaptive Clustering and Routing for wireless sensor networks

Cette section présente FLoR, notre algorithme de routage adaptatif. la gestion de la mobilité et l'équilibrage de charge sont les principaux problèmes à relever dans les réseaux de capteurs ad-hoc.

Pour résoudre ces problèmes, FLoR a 5 parties principales : Clustering (élection à la tête de Cluster), la gestion de la mobilité, l'équilibrage de la charge, la gestion de l'indisponibilité et la décision de logique floue (figure *g*).

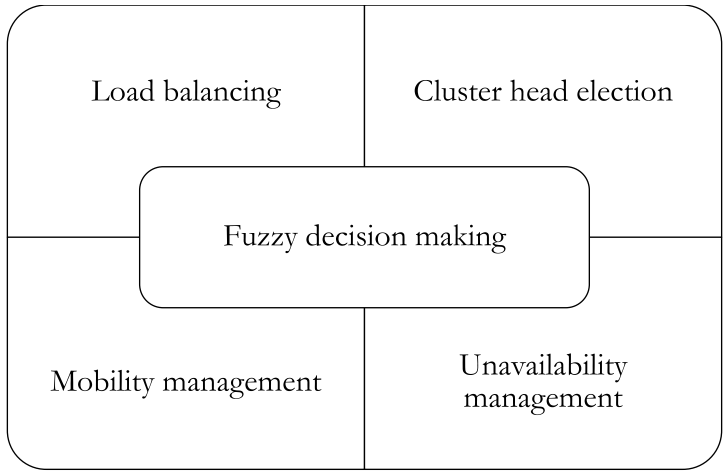


Figure g. Les parties principales de FLoR

a. La gestion de la mobilité et une nouvelle définition de « mobilité »

Afin de gérer la mobilité, nous avons proposé un nouveau paramètre appelé « mobilité ». Ce paramètre indique la fréquence de changement de parent. Donc chaque fois que le nœud change de parent, il doit incrémenter la valeur d'une variable. Il est clair que la mobilité d'un nœud fixe peut être supérieure à zéro, en raison de la mobilité de son parent.

b. Équilibrage de charge : notre définition de la charge

Notre stratégie d'équilibrage considère la charge cumulative dans un arbre de charge sur leurs nœuds parents. Nous utilisons l'arbre de charge (Load tree) et condition d'admission pour l'équilibrage de charge.

La figure *h* montre un exemple d'un arbre de charge. L'arbre de charge est enraciné dans la station de base. La charge est la moyenne de la QoL (voir la section suivante) entre un nœud et de ses enfants. C'est une nouvelle définition qui peut être utilisée comme un nouveau paramètre de QoS.

Dans un arbre de charge, le poids de chaque lien, dans l'arborescence de charge est QoL entre chaque nœud et son parent, et la charge de chaque nœud est la somme des QoLs entre le nœud et son enfant. Pour un nœud n, nous avons :

$$n.Laod = \frac{(\sum_{n.Child} n.Child.Load) + (\sum_{n.Link} n.Link.Value)}{\text{n. numberLowLevelCH} + \text{n. numberLowLevelLN}}$$

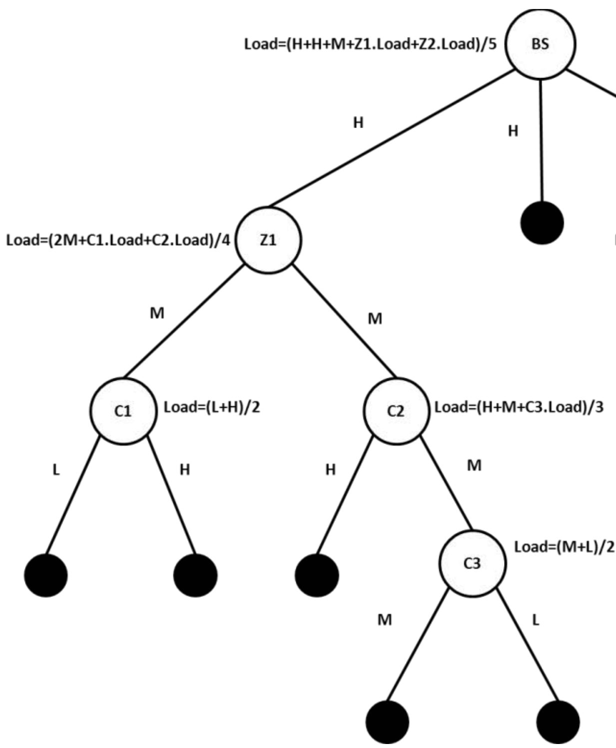


Figure h. A load tree

Afin d'équilibrer la charge entre les nœuds du réseau, nous utilisons la condition d'admission, qui est la condition d'accepter un nouveau nœud enfant au nœud n:

$$(\text{n.Load} + \text{NewChild.QoL}) / \text{n.numberLowLevelLN} + 1 \leq \text{n.QoL}$$

c. Gestion d'indisponibilité

Nous utilisons une technique pour améliorer notre algorithme de routage en considérant une histoire de l'indisponibilité des nœuds. En utilisant cette technique, notre algorithme de routage s’adapte dynamiquement à l’indisponibilité des nœuds.

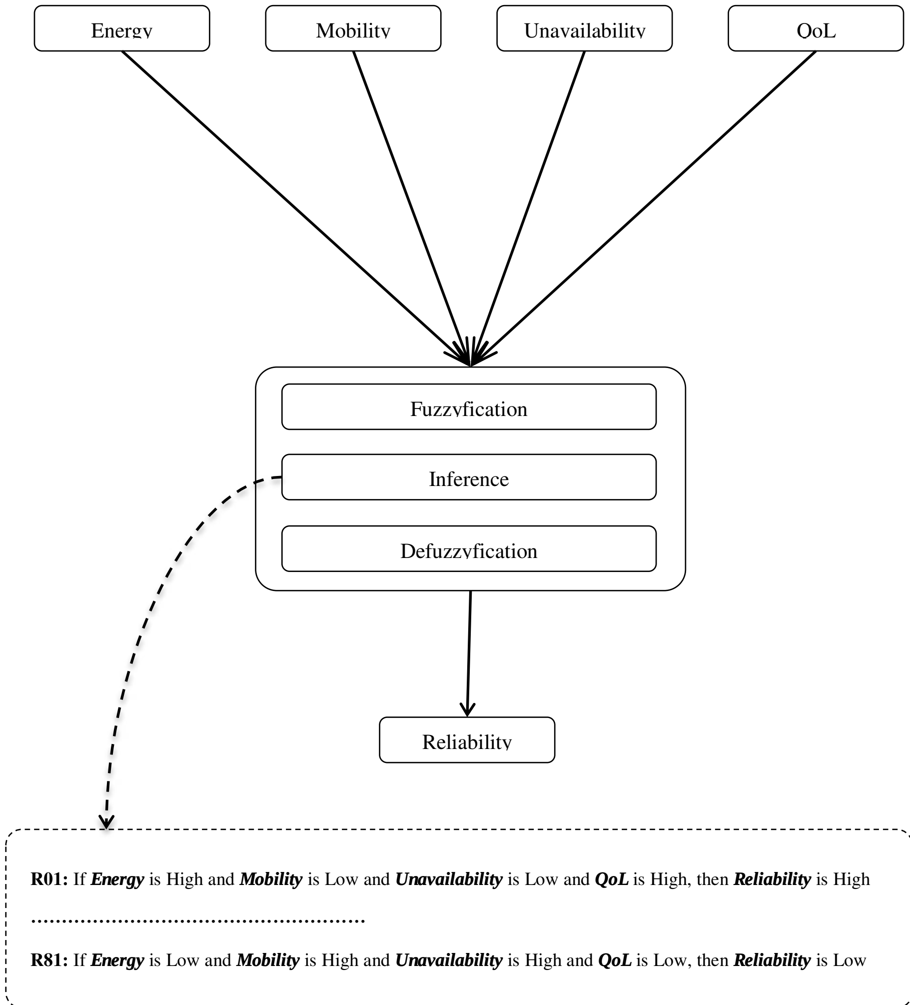


Figure i. Cluster head election

Pour ce faire, nous utilisons le paramètre Unavailability qui présente le taux de l’indisponibilité d'un nœud, détecté par son voisin. Ce paramètre sera calculé en utilisant le nombre d'indisponibilités du nœud au cours de sa durée de vie. C'est également une variable floue qui a 3 niveaux : High, Medium et Low.

Chaque fois que le nœud détecte une indisponibilité chez un voisin, il met à jour paramètre de la non-disponibilité de ce nœud :

$$Unavailability = fuzzy\ (n\ /\ L)$$

où fuzzy est une fonction pour convertir une valeur décimale à valeur floue, n est le nombre d'indisponibilités du voisin et L est la durée de de vie, donc le paramètre indisponibilité a une valeur différente pour chaque nœud dans les autres nœuds.

Dans chaque réseau, en raison de la mobilité ou le taux de l'indisponibilité des nœuds, Station de Base définit une période de mise à jour, à laquelle chaque nœud mettra à jour Unavailability, ce paramètre est donc vraiment dynamique qui peut changer non seulement de Low à High, mais aussi du High à Low.

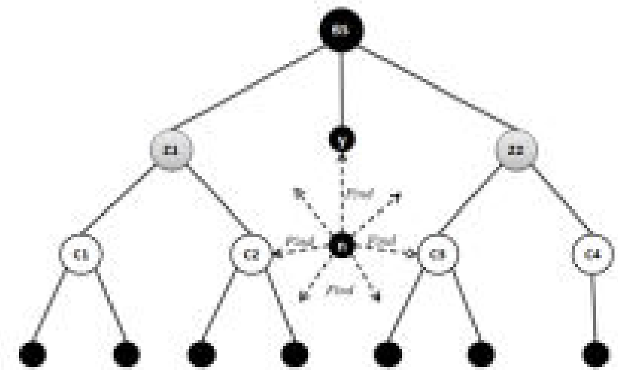
d. Comment fonctionne-t-il ?

Nous utilisons quatre paramètres : niveau d'énergie du nœud (charge de la batterie), mobilité, qualité du lien - QoL (fiabilité entre un nœud et son parent) et l'indisponibilité, pour évaluer un nœud qui est candidat à être un « cluster head ».

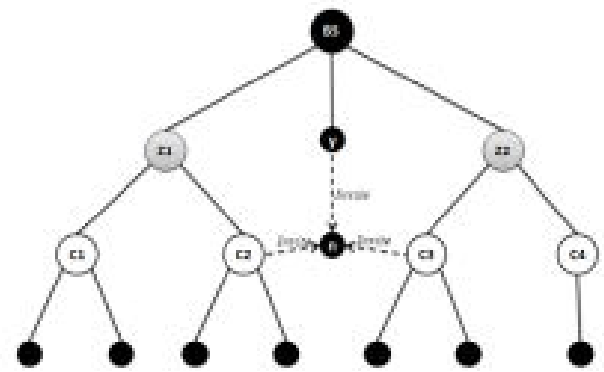
Ces paramètres seront les descripteurs de logique floue et chacun d'eux a trois valeurs possibles : High, Medium, Low. Par conséquent, nous avons des 81 règles maximum pour évaluer un nœud. Le résultat des règles sera la fiabilité avec 3 niveaux possibles : High, Medium, Low (Figure *i*).

Tableau 2 montre le premier exemple. Dans ce réseau le nœud n recherche un parent. Voici les différentes étapes de la procédure.

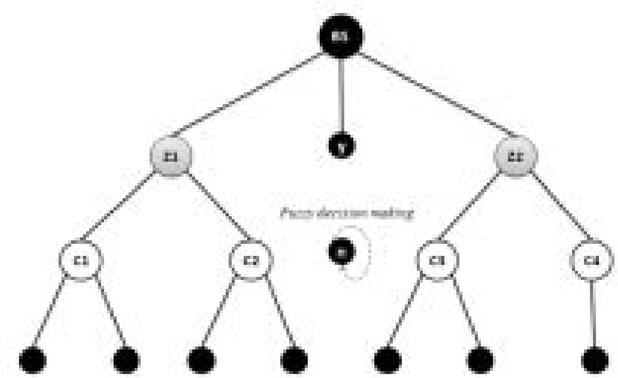
Tableau 2. UN EXEMPLE



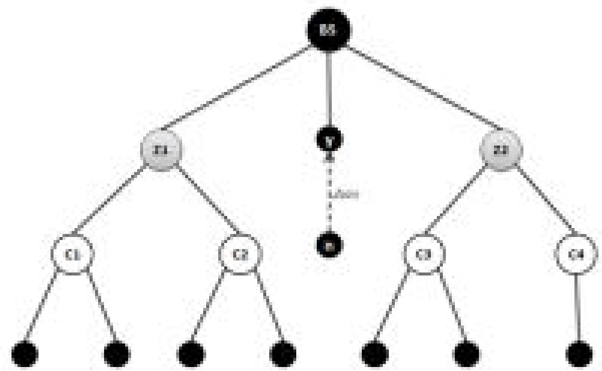
a. n diffuses Find



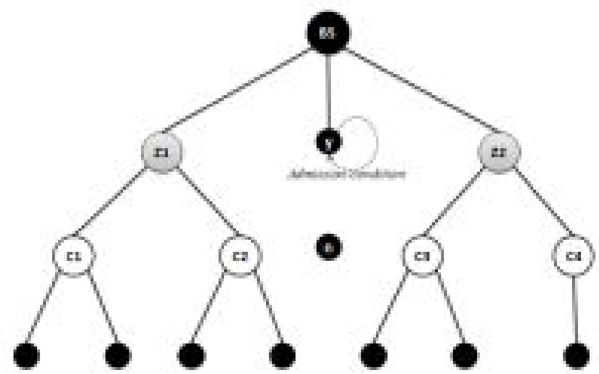
b. Son voisin, y, C2 et C3 lui répondent par Invite. Par ce message, ils envoient leur Energy, QoL.



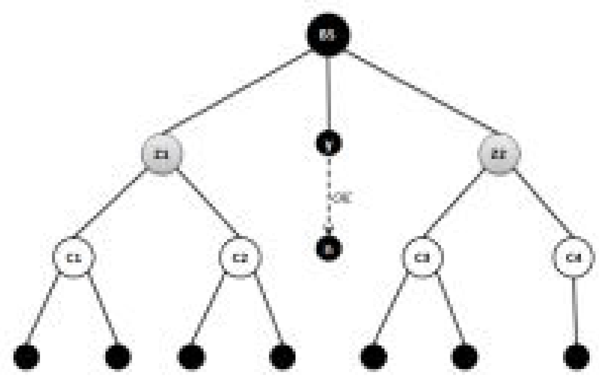
c. n exécute une fonction Fuzzy afin d'évaluer les candidates.



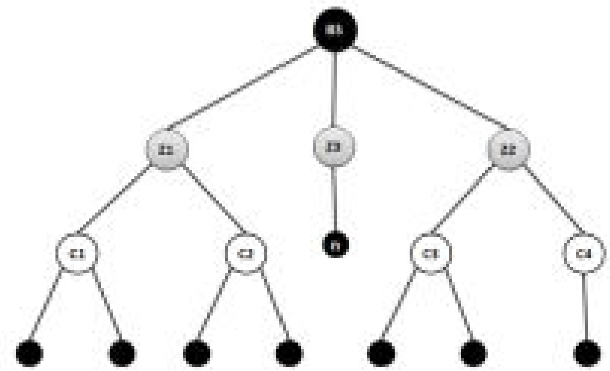
d. y est le meilleur et donc, n envoi Join.



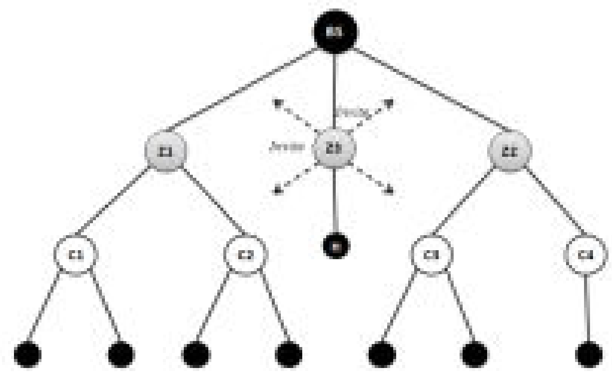
e. En recevant le message Join, y vérifie la condition d'admission.



f. La condition d'admission est ok, donc y envoi un message OK message to n.



g. n rejoint y, et l'état de y sera changé pour Zone Head.



h. Comme l'état de y est changée, y diffuse un message Hello à l'annoncer.

iv. Evaluation

Cette section présente l'évaluation de la performance d'FLoR. Nous avons comparé l'FLoR avec AODV [54] et LEACH [35]. Nous avons utilisé OPNET pour évaluer les FLoR. Le Tableau 3, montre les paramètres de notre environnement de simulation.

Les résultats montrent que notre protocole s'adapte grandement à la mobilité et à l'indisponibilité des nœuds.

Nous avons observé en LEACH, que le Throughput est inférieur aux autres mais qu'il augmente avec la taille du réseau. FLoR montre un meilleur rendement que les autres. En AODV, le résultat d'Overhead est en augmentation sur certains points et diminue à d'autres points comme la vitesse de transmission de paquets. Dans LEACH, l'Overhead de routage diminue régulièrement quand le nombre de nœuds augmente. Par rapport aux autres deux protocoles, la performance d'FLoR est meilleure avec plus de nœuds.

En ce qui concerne de délai, dans l'FLoR, le retard moyen de bout-en-bout devient très faible et augmente lorsqu'on augmente le nombre de nœuds, mais lentement et mieux que les autres. Pour la consommation d'énergie FLoR réduit la consommation d'énergie des nœuds lorsque le réseau a 75 et plus de nœuds. Lorsque le réseau est composé de 25 capteurs, LEACH a la plus faible consommation d'énergie. Dans un réseau avec 50 nœuds, LEACH a la meilleure consommation d'énergie.

La simulation montre qu'en augmentant le nombre de nœuds du réseau, la consommation d'énergie d'FLoR diminue, en revanche dans LEACH et AODV la consommation énergétique diminue en augmentant le nombre de nœuds du réseau.

Tableau 3. PARAMETERS DE LA SIMULATION

Node Number	25, 50, 75, 100
Surface	500m x 500m
Transmission range	15m
Data transmission rate	15 packet/sec
Failure model	Random
Packet size	128 bytes
Initial Energy	5J
Energy consumption (Calculation, receive and send)	10 nJ/bit

v. Discussion

Dans ce chapitre, FLoR, un algorithme de routage adaptatif a été proposé. FLoR utilise la logique floue pour former le cluster dynamiquement (voir la figure *j*). Les clusters sont formés dynamiquement pour s'adapter à l'évolution du réseau. Tous les processus décisionnels sont basés sur la logique floue.

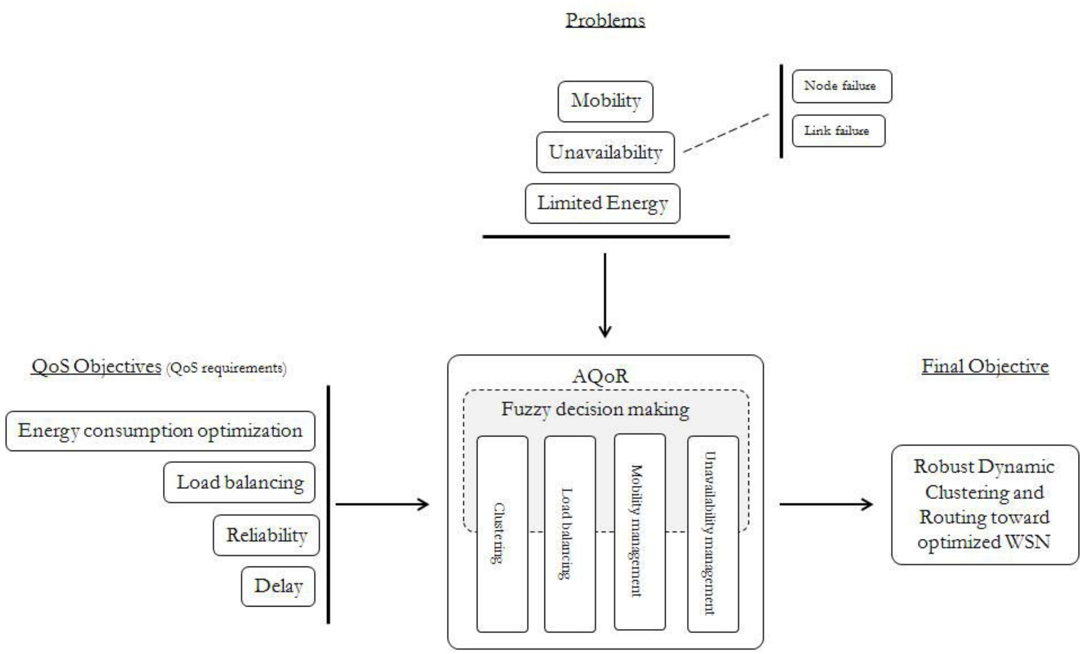


Figure j. FLoR

Afin de fournir un réseau dynamique robuste en tenant compte des objectifs de la QoS, FLoR utilise l'énergie, la mobilité, histoire d'indisponibilité et charge des nœuds qui sont de nouveaux concepts qui ont été introduits dans cet algorithme

C. Sensibilité au contexte et QoS adaptative [10]

L'expérience acquise au cours des 2 premières années de la thèse nous a montré la nécessité d'un pont entre les applications et les réseaux de capteurs pour garantir:

- La QoS dans le système
- La supervision du système
- La perception de l'état réel de l'utilisateur et de son environnement
- La configuration et l’installation des capteurs et actionneurs
- Le masquage de l'hétérogénéité et de la complexité du système et des sous-systèmes pour les développeurs d'applications et les utilisateurs finaux.

Pour ces raisons, dans cette thèse, nous avons développé CodaQ, un intergiciel (middleware) sensible au contexte permettant la gestion en-ligne de la qualité de service pour une communication fiable et optimisée (e.g. respect de contraintes temps réel, de fiabilité sous contrainte d'énergie).

L'objectif est d'aboutir à un système de communication (sans fil pour la plupart) robuste auto-organisé et auto-réparable intégrant différents protocoles selon les besoins en débit et selon les contraintes applicatives et protocolaires (Zigbee, Wifi, protocoles domotiques, ...).

i. Introduction

Le suivi d'activité (ou actimétrie) des personnes âgées à la maison est l'une des applications importantes des systèmes de télésoins. L’Actimétrie a besoin des capteurs en temps réel et donc d’un système de communication efficace avec la garantie de QoS.

Un Réseau de capteurs sans fil est une technologie émergente pour le renforcement des systèmes de télésoins et a incité les efforts de recherche importants. L'un des principaux thèmes de recherche est de fournir la QoS en termes de rapidité et de fiabilité en prenant en compte des contraintes énergétiques strictes. Cependant le fait de mapper les besoins de l'application et la qualité de service de réseau et encore un défi très important dans ce domaine.

Dans un système d'actimétrie lex flux de données dans le système peuvent être classés en 3 catégories avec des priorités différentes:

- Médical;
- Capteurs de l'environnement;
- Multimédia.

Le flux de données médicales possède la plus haute priorité et le flux multimédia le plus bas.

La collecte des données peut être réalisée par une passerelle domestique. Différentes stratégies de collecte de données aussi peuvent être appliquées dans chaque catégorie:

- basé sur des événements ;
- la collecte périodique de données
- le méthode mixte,

qui demandent la gestion des exigences de qualité de service (en appliquant les priorités) et de la cohérence temporelle (horodatage).

Les exigences de qualité de service des données du capteur sont déterminées par le type de données du capteur et l'état du patient. Par exemple, les données médicales comme l'ECG peuvent exiger une plus grande priorité que les autres capteurs quand un patient a un rythme cardiaque anormal. Dans ce cas, la passerelle domestique doit être en mesure de donner la plus grande priorité au flux de données ECG.

La cohérence temporelle est nécessaire pour la fusion multi-capteur de données (par exemple les données contextuelles sont recueillies pendant une fenêtre temporelle) et ceci peut être réalisé par l'horodatage à la passerelle domestique.

En outre les données des capteurs doivent être interprétées dans leur contexte actuel. Ceci peut être réalisé en utilisant le calcul de la sensibilité au contexte. Le contexte est toute information sur l'utilisateur et l'environnement qui peut être utilisée pour améliorer l'expérience de l'utilisateur.

Les systèmes sensibles au contexte représentent des systèmes distribués extrêmement complexes et hétérogènes, composés de capteurs, des actionneurs, des composants d'application, et une variété de composants de traitement / actionneurs et les applications. L'objectif de notre travail est d'en déduire l'état du patient à tout moment et les informations contextuelles nous aident à le faire.

ii. Le modèle référence

Pour bénéficier pleinement des avantages des systèmes sensibles au contexte, une infrastructure qui permet une auto-adaptation de QoS et l'interopérabilité entre les composants hétérogènes est nécessaire. Dans ces systèmes, tous les changements de contexte doivent être détectés rapidement et l'adaptation de système à des nouvelles conditions de contexte doit se fait de manière automatique et rapide. Ceci est important dans les systèmes de télésoins qui surveillent une personne avec des problèmes de santé.

Afin de répondre aux besoins et adresser les défis dont nous avons parlé, nous avons défini un modèle de référence (Figure *k*).

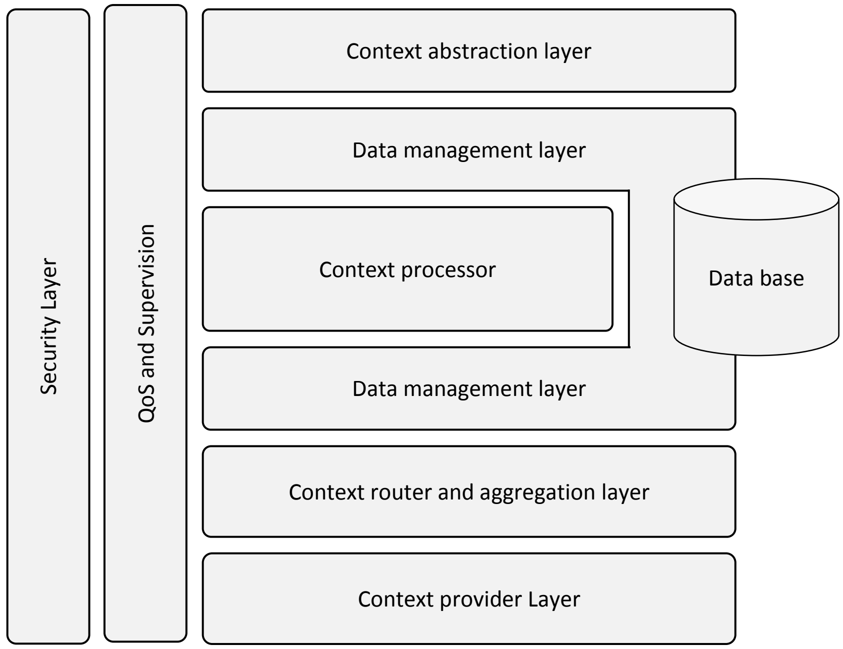


Figure k. Reference model

Ce modèle présente tous les modules que doit avoir un middleware, pour être en mesure de faire face aux défis et d'assurer une déduction performante de contexte. Dans ce modèle:

- Context abstraction: fournit une abstraction du contexte pour les applications sensibles au contexte.
- QoS and Supervision: assure la surveillance du système et s'applique les besoins de QoS.
- Database: enregistre les événements et les états déduits à partir du contexte.
- Context processor: utilise les données brutes fournies par le "Context router", pour déduire l'état réel du contexte.
- Data management: contrôle les bases de données et assure la gestion de toutes les données qui sont échangées entre les autres couches.

- Context router and aggregation: est une passerelle entre les capteurs et le middleware.
- Context provider: est une application distribuée installé sur les capteurs pour détecter et envoyer les événements au middleware.
- Security: est la responsabilité de fournir la sécurité du système.

Dans cette thèse nous présentons CodaQ (A Context-aware and Adaptive QoS-aware Middleware for Activity Monitoring) un middleware contextuel, qui est un élément clé de notre système de télésoins. CodaQ utilise la logique floue et traite un grand sous-ensemble des conditions énumérées dans le tableau 1, sauf la sécurité et le « privacy » qui sont nos futurs travaux.

iii. Modélisation de données dans CodaQ

Dans CodaQ toutes les données obéissent à une représentation uniforme à l'aide de la modélisation des données. Les variables/attributs utilisés pour la modélisation sont :

L'événement brut noté par **RawEvent.** C'est la sortie d'un capteur sans aucun traitement; Il montre un événement découvert par un capteur dans un endroit spécifique (Zone d'un habitat), comme le mouvement ou la présence d'une personne, ouverture et fermeture de la fenêtre ou la porte etc. L'Événement **RawEvent** est défini par :

RawEvent (Id, Time, SensorID, {Values}, Zone)

Le capteur virtuel (Figure *l*) est la composante principale d'abstraction du matériel dans MPIGate. Le descripteur de capteur virtuel dans MPIGate est défini par : **VirtualSensor**(Id, {Sensors})

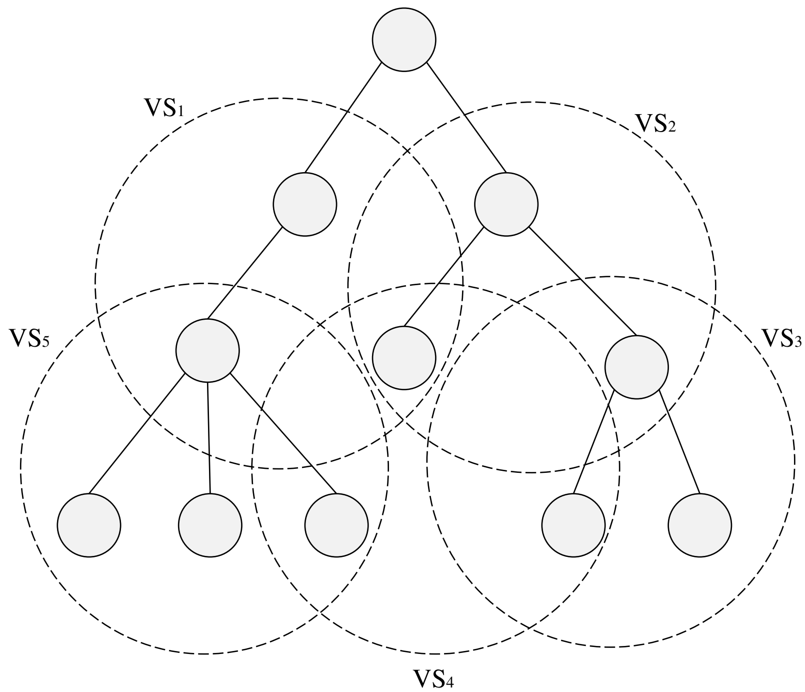


Figure 1. Capteurs virtuel

L'entrée d'un capteur virtuel est un évènement brut et sa sortie est un état déduit. L'état déduit est le résultat des processus sur les événements bruts et présente la situation réelle du contexte. L'état déduit est représenté par **DeducedState**:

DeducedState (Id, Value, Category, Time, Zone/Place, VirtualSensorId)

Pour un utilisateur, la valeur de **DeducedState** pourrait être : la Chute, Partir/Arriver, Dormir, Manger, etc.

La zone, présente l'endroit d'un événement donné ou d'un état :

Zone (Id, {Sensors})

La requête est l'ordre produit par le système pour lire la valeur d'un capteur ou commander un actionneur. La requête est représentée par le descripteur:

Query (Id, Category, {Sensors}, {Actuators}, Zone)

iv. Architecture de CodaQ

Dans l'architecture de CodaQ (Figure *m*) c'est *Context Collector* (équivalent de *Context Router* du modèle de référence) qui recueille le contexte des événements premiers auprès des fournisseurs de contexte distribué et les enregistre dans la base de données *Raw Event* en passant par *Data Management*. Cette couche a une sous-couche appelée *Queue Manager* qui contrôle le buffer d'entrée par des informations reçues de la part de *QoS Specification Provider*.

La couche suivante est la couche *Data Management*, qui possède 2 sous-couches:

- *Data Manager*: reçoit les requêtes des différentes composantes de l'intergiciel, les exécute et renvoie les résultats.
- *Database*: se compose de six bases de données.

Couche *Context Processor* déduit l'état réel du contexte à partir des événements bruts en utilisant des techniques de raisonnement fondé sur des règles de logique floue. En outre, la validation temporelle et spatiale doit être prise en compte aussi dans cette couche.

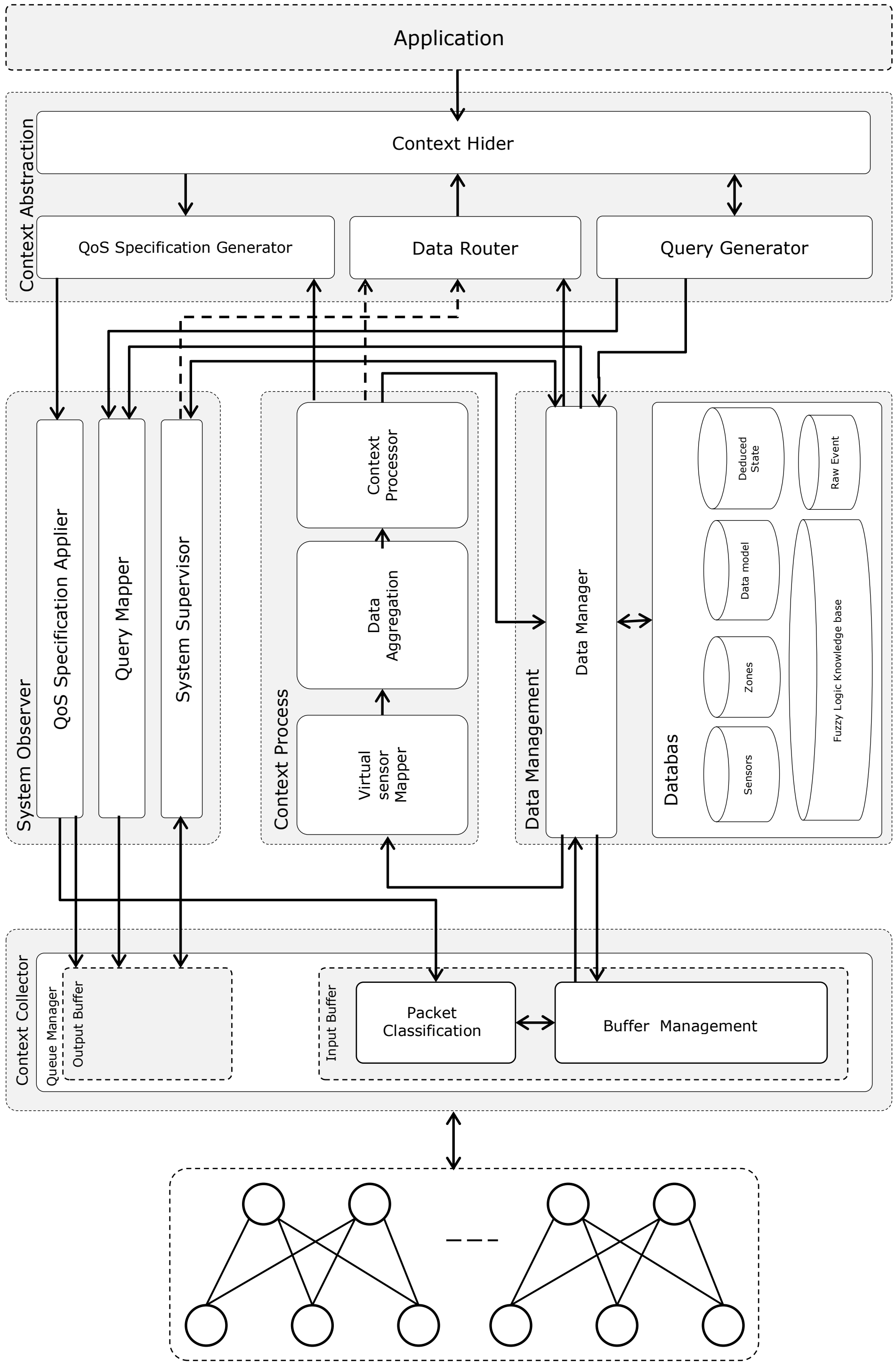


Figure m. Architecture de CodaQ

Pour tenir en compte de ces besoins, cette couche a trois sous-couches:

- *Data aggregation:* regroupe les informations de contexte à l'aide de données brutes, des informations fournies par Virtual Sensor et des transferts à *Context Processor*. Cette couche vérifie aussi la validité temporelle dans une fenêtre temporelle donnée. Si les données d'un ou plusieurs ne sont pas dans la fenêtre de temps, cette couche demande à *Query Generator* d'envoyer une requête aux capteurs afin de fournir les données fraiches.
- *Virtual Sensor Mapper*: fournit les informations nécessaires sur tous les capteurs virtuels du réseau. Il utilise des bases de données *Sensor* et *Zone* pour créer des capteurs virtuels et leur fournit à la couche *Data aggregarion*. Le concept de capteur virtuel est utilisé pour appliquer la validité spatiale.
- *Contexte processor*: Utilise les informations fournies par *Context aggregation* et les processus afin d'en déduire l'état réel du contexte.

Couche QoS Observer est la couche suivante de CodaQ qui a trois sous-couches:

- *QoS Specification Applier*: dans notre conception la qualité de service QoS s'applique à deux niveaux: *Embedded* et *Run-time* QoS. Pour la garantie *Embedded QoS*, nous utilisons *FLoR*, notre algorithme de routage adaptatif (Chapitre 3) qui optimise l'utilisation des ressources en tenant compte de l'énergie, de la mobilité, de l'histoire de défaillance et de charge des nœuds. Le *Run-time* QoS est basé sur les priorités qui sont définies par l'application dans un temps donné, afin d’assurer les besoins d’application en terme de ressources demandées.

- *System Supervision*: il crée des requêtes et les envoie vers les destinations précisées (capteurs et actionneurs) afin de surveiller l'état des capteurs et des actionneurs installés et de détecter toute défaillance dans les nœuds ou les liens de communication.
- *Query Mapper*: utilise les bases de données *Zone* et *Sensor* pour générer les requêtes nécessaires afin de fournir les données fraiches des capteurs.

Enfin, la couche *Context abstraction* uniformise les informations et les données transférées entre différentes parties d’application et le middleware en appliquant les définitions de la modélisation des données que nous avons définies. Il se compose de trois sous-couches:

- *Context Hider*: reçoit les requêtes de l'application et les envoie à *QoS Specification Generator*. Il reçoit également les résultats des requêtes et les envoie à la couche application.
- *Query Generator*: Cette sous-couche est le pont de la communication entre la couche application, les capteurs et les actionneurs et permet aux applications d'extraire des informations de contexte désirées à partir des réseaux.
- *QoS Specification Generator*: génère les spécifications de QdS en prenant en compte des besoins et la demande d’application.

D. Autres activités

Nous avons participé au concours de ministère de l'Enseignement supérieur et de la recherche avec un projet intitulé MPIGate : « Multi Protocol Interface and Gateway for Telecare, Environment Monitoring and Control » et nous avons été lauréat de ce concours en catégorie « émergence ».

Le projet MPIGate est issu de divers travaux de recherche dans cette thèse au sein de l'équipe TRIO dans le domaine des réseaux personnels sans fil (WPAN), et en particulier dans celui des réseaux de capteurs sans fil pour des applications de téléassistance, maintien à domicile, domotiques, industrielles et de logistique.

L'objectif de ce projet est d'accompagner les entreprises du domaine de la télésanté, de l'aide à domicile et de l'industrie dans leur évolution vers la technologie des WPAN. La MPIGate est une passerelle facilitant la communication entre les WPAN et d'autres technologies filaires et non filaires. Elle inclut une interface "opérateur" et une interface modulaire "utilisateur" qui sera adaptée aux besoins des applications envisagées.

Le MPIGate a pour objectif d'offrir une passerelle entre plusieurs technologies d'acquisition, d'actionneur et de communication. Nous proposons pour cela une architecture générale découpée en plusieurs modules indépendants dont certains sont extensibles. Cette architecture (Figure *n*) permet de faire interagir plusieurs éléments technologiquement hétérogènes de façon robuste. L'architecture se répartie en cinq parties :

- Le module d'administration: Il permet de gérer les objets de données, la configuration de la connectivité, le profil des extensions du collecteur et de l'émetteur.

- Le module de connectivité: Il permet la communication avec des logiciels ou services extérieurs au MPIGate et ce de façon sécurisé.
- L’espace d’objets Il contient une représentation de chaque entité définie par l’émetteur et le récepteur.

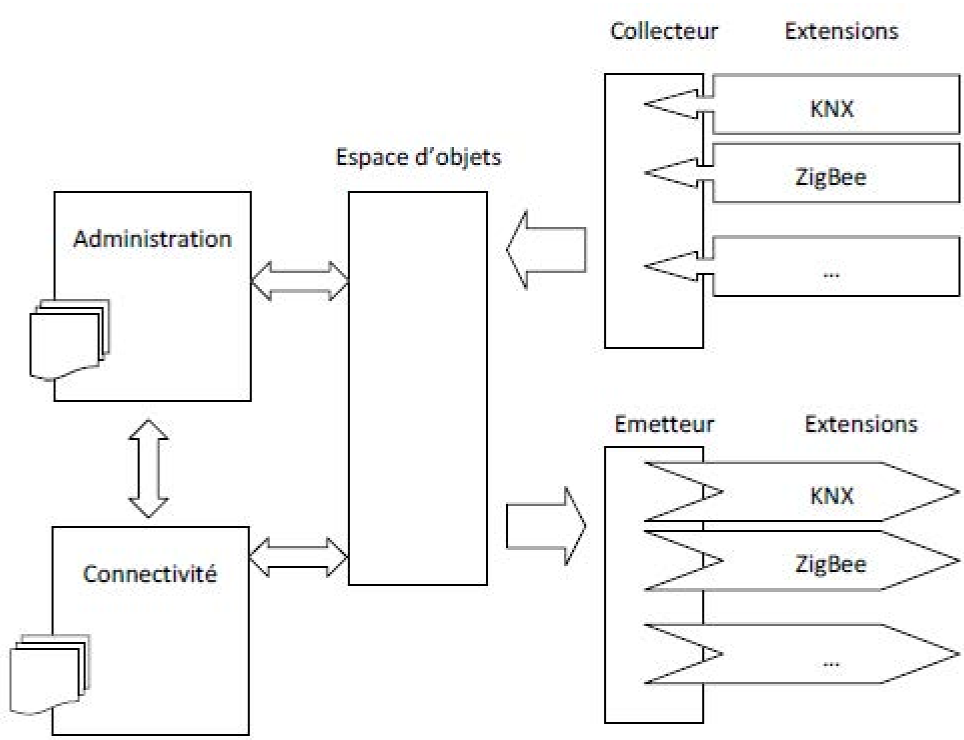


Figure n. Architecture générale

V. Implémentation

Ce chapitre décrit la mise en œuvre et l'approche de développement qui a été suivie pour évaluer notre système sensible au contexte et l'architecture middleware qui le soutient. Un prototype a été mis en œuvre pour vérifier la faisabilité de nos propositions, fondées sur notre appartement de laboratoire à Colmar.

L'environnement de programmation de l'intergiciel est Visual Studio .Net 2005 C # et ASP. Net et la découverte de service est basée sur SOAP. Le prototype utilise une base de données SQL Server. Le prototype utilise des capteurs Imote2 dans la partie réseau de capteurs. L'environnement de programmation pour ces capteurs est Imote2.Biulder SDK sous Visual Studio. Net 2005 C # et Microsoft. Net Micro Framework 2.5, basé sur TinyOS 2.0.

Nous avons examiné le temps de réponse d'une requête, qui est défini comme l'intervalle entre le moment où la requête est émise à partir d'une demande présentée par la réception des données du capteur et le moment où la demande reçoit le résultat de la requête comme un retour de l'actionneur.

Dans ce scénario, nous utilisons motes Imote2 au sens de luminosité extérieure. Les données captées seront envoyés à la passerelle, et seront traitées par le middleware, et basé sur le résultat, la luminosité intérieure aura varié.

Le tableau 4 présente la répartition du temps d'une requête dans notre système. Les résultats présentés dans ce tableau sont la moyenne des 10 mesures. Le temps de réponse

Tache	Temps (***ms***)	%
Context collection	378	34%
Context processing	112	14%
Query generation	34	4%
Reply	237	48%
Total time	**797**	

La collecte des données est le temps passé sur l'obtention des valeurs d'attribut cadre du réseau de Imotes, le traitement contexte est le temps du traitement le contexte nécessaire pour évaluer les données et générer la requête, génération de requête est le temps passé par le système sur la création de la requête et enfin est la réponse du temps nécessaire pour envoyer le résultat de la requête à la demande.

Dans le tableau, nous voyons que, le temps de réponse total d'une requête dans notre système est inférieur à une seconde. La collecte de données de contexte à partir du réseau de capteurs prend 34% du temps total et la tâche la plus fastidieuse parmi les autres est le délai de réponse qui prend 48% du temps total.

Le temps de traitement du contexte, prend 14% et le temps de génération de requête est seulement équivalent à 4% du temps de réponse total. Si nous utilisons la commande directe d'un actionneur nous n'avons pas besoin de collecter le contexte.

Cela signifie que par l'automatisation de l'environnement à domicile par notre système, nous ajoutons une surcharge de 273 ms pour la collecte de contexte et 14% pour la transformation du contexte ou 48% du temps de réponse total qui semble être acceptable que le coût de l'automatisation du traitement et de contexte de contrôle.

VI. Les travaux futurs et la discussion finale

Nos expériences et nos développements ont donné naissance à un certain nombre de questions intéressantes à atteindre dans la poursuite des travaux. La prochaine étape, qui a déjà commencé, est de tester le système dans un environnement de monde réel d'un appartement intelligent au Loria.

À cet égard, avant de promouvoir un système final, la méthodologie de spécifications, de développement et d'intégration des caractéristiques technologiques doit être réalisée comme avec une préoccupation majeure pour certains concepts importants,: l'interopérabilité, évolutivité, évolution et personnalisation selon les profils médicaux et sociaux et les besoins des utilisateurs. En développant cette plate-forme expérimentale, nous avons les objectifs suivants :

- Recherche et développement d'un intergiciel interopérable pour un système d'accueil en réseau qui combine la domotique, réseaux de capteurs sans fil (ZigBee, 6Lowpan…), mobile et des fonctionnalités de PC qui peuvent être redimensionnées en haut et en bas pour l'utiliser sur n'importe quelle plate-forme matérielle et logicielle dans ces domaines.
- Recherche et développement de services d'utilisateur intelligent qui combinent l'interaction utilisateur, la préférence de l'utilisateur et la sensibilisation au contexte.
- D'interopérabilité garantie au sein de ce système à domicile entre services même avec l'utilisation de réseaux hétérogène et divers dispositifs.
- Garantie dynamique configuration automatique du système home en réseau en développant des méthodes de haut niveau et de concepts pour intégration dynamique traitant des aspects d'autonomie et d'évolutivité.
- Vérifier les solutions techniques avec les prototypes d'application et les manifestants connexes et obtenir la rétroaction de l'utilisabilité. Les prototypes mettront l'accent sur les soins à domicile et de sécurité, de domotique et de l'environnement familial prolongée.

Basé sur cette plate-forme expérimentale, une autre œuvre future comprend une extension de notre technique de fournir un contexte efficace afin d'offrir un soutien important pour middleware sensibles au contexte dans différents domaines (sensibilisation de contexte, de raisonnement, de négociation, de personnalisation et d'adaptabilité pour les services de traitement et la connectivité de réseau).

À côté du fournisseur de contexte, nous visons à définir un algorithme préventif de gestion de routage et de ressources dans les réseaux de capteurs et actionneurs basé sur une modélisation comportementale de la personne.

Une œuvre future, qui est déjà commencée (MPIGate), est de développer une passerelle multiprotocole. L'objectif est de permettre la communication entre les capteurs hétérogènes et robots sans ou avec les efforts d'un minimum de configuration manuelle.

À côté de traitement de contexte, nous examinerons la caractérisation temporelle des activités. Dans ce but, nous prévoyons de concevoir une extension de notre définition de modèle de données à une définition de l'ontologie et d'enquêter sur l'utilisation prolongée du système d'envisager une classe plus large des données de contexte. En outre, étant donné que dans de nombreuses situations données disponibles contexte peuvent être insuffisantes pour déterminer sans ambiguïté l'activité effectuée par un utilisateur, nous étudions l'utilisation des ontologies floues pour gérer les incertitudes et flou.

Chapter 1 Introduction

Nowadays most of dependent elderly requiring healthcare services must leave their home and move to special healthcare centers. However this is often not well accepted for them because of high cost and isolation from their family. Advanced communication networks and Internet technologies play a key role in the development of smart homes for AAL (Ambient Assisted Living). Healthcare integrated smart homes can be used to allow the seniors and their family members to benefit from a safe home environment with activities and services accompaniment for their daily life. To reach these benefits, most countries are now facing an urgent requirement to provide appropriate retired home environment solutions for senior citizens and allow them to stay at their own homes. For the same reasons, elderly telehomecare, videoconferencing and tele-consultation systems have been attempted by many researchers, but most of them provide only partial solutions for specific case studies [12]. So we aimed to propose a general, modular and scalable telehomecare and activity monitoring system, by considering important technical and scientific challenges such as QoS (Quality of service), interoperability and easy data access for developing intelligent AAL.

1.1 Context

As the population grows older, people becomes increasingly dependent as their sensory, motor and cognitive physiological health capacities deteriorate; these age-related changes are amplified if they are accompanied by pathological conditions (e.g. osteoporosis, osteoarthritis) that are common in the elderly population. Most European countries are now facing an urgent requirement to provide appropriate retired home environments solutions for these citizens and allow them to continue to play a role in our society. Today, elderly requiring healthcare services must move to distant medical centers and this is often not feasible due to their social and economic stares. Elderly telehomecare systems can be achieved using advanced communication networks and Internet technologies. These technologies have a key role in the development of a telehomecare system and smart homes.

Healthcare integrated smart home allows the tenants and their family members to benefit from a safe home environment with activities and services accompaniment for their daily steps. In order to reach that goal, we aimed at several complementary technologies and implemented them in a demonstration prototype apartment [1-5]. Several issues on objectives and contributions of technologies were stressed out:

- Enhance independent living with cognitive assistive systems.
- Provide safety and security to inhabitants with the help of medical services supervision; monitor general activities of the occupants and monitor unusual or abnormal behaviors that reach dangerous levels.
- Improve social contact and cultural activities within the elderly surrounding thanks to communication systems and appropriate equipment.

- Engage medical actors to use computerized systems in order to ease information processing and traceability.

For these reasons, elderly telehomecare, activity monitoring [11], videoconferencing and tele-consultation systems have been attempted by many researchers. Many telemedicine and e-health systems are being developed and many innovations and ICT-based emerging solutions are close to be operational, nevertheless the expected take-up did not occur yet, since proposed systems and their targeted medical benefits are certainly too segmented and disconnected the ones from the others. In addition, most of existing products and services in this field rely on proprietary technologies, which is a substantial obstacle to the development of complementary services or applications, and thus of the market. Consequently, population like elderly, disabled people or chronically ill people, who are often suffering from co-existing troubles, are not encouraged to dare benefiting from ICT and innovations, as available solutions are not complete, not compatible the ones with the others, not scalable, neither end-to-end nor comprehensive, and remain then unaffordable for someone who would try to subscribe to each relevant service covering a more or less significant part of its medical and social needs, without ensuring a global management of the daily problems. In summary, the current developments and innovations are limited in terms of commercial and operational perspectives and suffer from certain phenomena:

- The fragmentation of experimentations, projects and products did not yet give birth to any efficient and affordable holistic solution that could fulfill the various targeted beneficiaries and users' expectations [12].
- The development costs and R&D expenses required for a proper adaptation to each community of users and beneficiaries' needs make

investment very high, as time-to-market delays are uncertain and no business model has been successfully assessed yet in this domain.

- The accompanying measures, at institutional, ethical, legal, financial and organizational levels remain insufficient, which prevent the development and the sustainability of such assistance and at-home monitoring services.
- The population is neither confident nor familiar enough with these innovation-based solutions, especially in such a medical or health-related purpose, although their development requires that they become a natural component of the medical and social environment and infrastructure.

Fortunately, our society is being aware of this situation and many governmental initiatives have been made to encourage the development of enhanced research activities toward building operational systems and solutions. In Europe, call for AAL projects, and in France, "CNR Santé" (http://www.cnr-sante.fr/) and different recent calls for projects and the modifications made in telemedicine lows are some examples of these initiatives.

1.2 Challenges

In a telehomecare and activity monitoring system, several challenges need to be addressed. The technological challenges regard the generic features of the wireless sensor networks (e.g. mobility, failure, limited energy resource…), of the different supporting platforms, of the video call center, and web services, to allow a flexible and smooth interaction among those items and to anticipate further additional functionalities, interacting devices, and so on. In this respect, the methodology of specifications, development and integration of the technological features must be realized with a major concern for some important concepts, like: Context-awareness and self-adapting, Dynamic QoS (Quality of Service) support, Evolution and customization. In this thesis, we only focus on QoS, context-awareness and self-adaptation issues.

1.2.1 QoS support

As the healthcare industry is turning to information technology to help solve its business issues, specially provide to quality patient care services, it is important to develop QoS specification in distributed health information systems [1-5].

Unlike traditional systems, many non-end-to-end mission-critical applications envisioned for healthcare systems, which are complex systems, because in these systems, we can find different subsystems, such as, wireless sensor networks, LANs, software platforms, home automation systems, mobile systems, Internet… that require different QoS requirements on the system, and these requirements pose unprecedented challenges in the area of QoS support in healthcare systems.

Quality of service is an overused term with multiple meanings and perspectives from different research and technical communities. Conceptually, it can be regarded as the capability to provide assurance that the service requirements of applications can be satisfied. Depending on the type of target application, QoS can be characterized by reliability, timeliness, robustness, availability, and security, among others [13].

In wireless sensor networks (WSN), we can view QoS from two perspectives: *application-specific* and *network*. The former refers to QoS parameters specific to the application, such as sensor node measurement, deployment, and coverage and number of active sensor nodes. The latter refers to how the supporting communication network can meet application needs while efficiently using network resources such as bandwidth and power consumption [14].

Traditional QoS mechanisms used in wired networks aren't adequate for WSNs because of constraints such as resource limitations and dynamic topology. Therefore, in these networks new mechanisms must be provided to maintain QoS over an extended period and even adjust itself when the required QoS and the state of the application changes. This leads us to interest to develop adaptive QoS mechanisms.

QoS technique should be designed based on trade-offs among performance metrics such as network capacity or throughput, data delivery delay, and energy consumption. Many QoS parameters may be used to measure the degree of satisfaction of these services, but throughput, delay, and packet delivery rate are the most fundamental [15-17].

Energy consumption optimization is one of QoS challenges in wireless sensors. The wireless technology most used in the home networks is WiFi. But Wi-Fi consumes too much energy and drains the battery faster.

Due to this problem, there is advantage of using low-power consumption technologies like ZigBee, for which there is many energy consumption optimization mechanisms [92, 93].

Moreover, any failure or lack of performance on the system which could not be tackled in a reasonable delay may have some damageable consequences on the solutions' acceptance and development potential: the confidence is a basic and elementary factor of acceptance or reject, such incident could also generate a psychological defiance towards ICT's in general and towards such innovative assistance and monitoring services.

1.2.2 Context-awareness and self-adaptation

Healthcare systems present a growing demand for software applications that are able to adapt dynamically to changing environment and user conditions as well as to the unavailability of a devices or a services.

Context-aware and self-adapting [57, 58] systems are interesting solutions for this demand. The concept of adaptation and context awareness are now open research subjects as well as the application developments.

Context-aware healthcare is making healthcare services available everywhere, anytime by applying context-aware computing technologies for healthcare, health, and wellness management. Using context-aware computing device and network, the doctor would be informed of patients' context data such as vital sign, and would call his patients to encounter hospital for more accurate examination. To support the doctor for gathering and analyzing the patients' context-data, some intelligent systems are required.

In context aware computing environment, computing entities is ranging from sensors and actuators to web services and applications. For example, a wearable health monitoring device can constantly examine one's blood pressure, body temperature, pulse, etc.; the availability of surveillance camera and embedded microphones and home automation systems at home may support remote health and activity monitoring. To fully use the benefits of context-aware system, an infrastructure which enables QoS self-adaptation and protocol interoperability among heterogeneous components is required.

1.3 Thesis Contributions

This section briefly outlines the main contributions of this PhD work. Publications resulting from this dissertation are listed in the references [1-10].

1.3.1 "Maisons Vill'Âge®": A telehomecare system for elderly [1-5]

The presented requirements and challenges in 1.2 and remaining obstacles make the design of monitoring and assistance tools quite complex, as each pathology or disability generates its own set of requirements and constraints. Such context increases the need for the development of innovative global ICT-based solution allowing to implement personalized and person-centric care process and tools, which can evolve with the users' medical and social and physical evolution, and which can compile the various collection and analysis of data to ensure a reliable and sound diagnosis, medication and evaluation of the medical and social state of the person.

The first contribution of this thesis is the development of "Maisons Vill'Âge®", a new concept of building smart home by integrating telemedicine and home automation systems. We have designed and developed a distributed system architecture to support remote medical consultations and elderly management and homecare across global wide area networks and heterogeneous platforms. The system is designed for the elderly who wish to spend their old age at their own home, in order to increase independence and quality of life. This would not only benefit the elderly who want to live at their own home, but also the national health care system by cutting costs significantly. The first houses are being built with basic implements of data acquisition and human-machine interface. A flat is entirely equipped to act as demonstrator and test bed.

Further details can be found in chapters 2 and 5.

1.3.2 QoS support in wireless sensor networks [6-9]

In all of the data acquisition networks, like health monitoring systems, either the data is collected from the network periodically or on the occurrence of an event, in such systems, the data are highly vital to have a stable monitoring and a minimum number of faulty alerts. However, none of them adapts completely themselves neither to the failure of the nodes nor the temporal variations in data delivered by the sensor network.

In addition, limited energy resources in wireless sensor networks and unreliable radio channel, unavailability of the nodes is a constraint to provide a reliable communication. The unavailability of a node may have many reasons like mobility of the node and node or link failure. It is clear that in a real environment, we cannot control or reduce the number of unavailability in a network but it's possible to manage it. This necessitates the use of a routing algorithm with a QoS support mechanism which readily adapts to the unavailability and mobility of the nodes and changes in the data delivery rate.

The second contribution of this PhD work is the proposition of a fuzzy logic based QoS-aware routing algorithm is proposed. Stable route recovery, high data delivery ratio and good load balancing are the main characteristics that the proposed algorithm adds to the ad-hoc sensor networks. The algorithm uses a multicriteria decision to form the clusters and manage the mobility and unavailability of the nodes and because of this multicriteria decision strategy it uses fuzzy logic to facilitate the decision process.

This protocol adapts also with mobility and unavailability of the nodes. It is especially effective in networks that use sensor nodes for data aggregation and in which the data delivery ratio is important and the nodes are mobile, like health monitoring sensor networks. In such networks health events and information are sensed by several nodes and therefore, this protocol can help the network to deliver sensed events with high successful rate in the network. The simulation results show that the proposed protocol is well suited for such applications.

1.3.3 Context-aware adaptive QoS Middleware [10]

The experience gained through the research and development activities during the 2 first years of the thesis and our propositions for QoS routing algorithm and the healthcare system, showed us the lack of a link between the healthcare applications and the sensor networks. So a bridge is needed to guarantee:

- QoS needs in whole system
- Supervision of the system
- Deduction of real state of the user and his environment
- Configuration and installation of the sensor and actuators
- Hiding the heterogeneity and complexity of the system and sub systems to the application developers and end-users.

For these reasons, we developed, as our third contribution, a context-aware middleware, CodaQ, which has a new architecture. Context Collector, Data management, Context process, System observer, Context abstraction are the main components of the architecture of CodaQ. In this middleware all data are given a uniform representation, and their level of abstraction is raised through the use of data model definitions. A context-based adaptive QoS method was implemented on the system, in two levels: Embedded and State-based QoS.

1.4 Thesis Outline

The organization of the rest of this dissertation is as follows. Chapter 2 presents the architecture of our telehomecare system. In chapter 3 we discuss motivation, design, and performance of our adaptive routing algorithm for multi-hops wireless ad hoc sensor networks. In chapter 4 we present the motivation and design of CodaQ middleware. Then we present the implementation of our system in chapter 5. Conclusion and some future work are given in chapter 6.

Chapter 2 "Maisons Vill'Âge®"

Smart Use of Sensor Networks for Healthy Aging

The recent advances of ICT allow developing new generation of telehomecare systems to enable elderly health monitoring and management and ease their daily life. Such a system can reduce the amount of admissions to hospital, facilitate more efficient clinical visits, and may replace a hospital stay by living at home. In this chapter we present a patient oriented distributed telehomecare architecture, based on which we developed "Maisons Vill'Âge®", a concept for building smart home by integrating telemedicine and home automation systems.

Some preliminary results on system performance and acceptability are also analyzed, allowing revealing two important issues that merit future investigation: Fault management in QoS and easy context-aware data access.

2.1 Introduction

Technologies exist to help people deal with a reduction or loss of mobility, vision, hearing, and cognitive ability; to continuously monitor vital parameters; to reduce accidents by anticipating risky situations; and to deliver therapy through wearable biomedical sensors.

In this respect, smart homes are being used to maintain safety and independence among elderly and to provide telehomecare (home-based eHealth) services. The concept of telehomecare that includes both telehomecare and the smart home is introduced in [19]. Telehomecare refers to hidden illness and trouble prevention and monitoring of residents who may not receive other forms of home care, such as the disabled or elderly [19].

The smart home is a promising and cost-effective way of improving access to home care for the elderly and disabled. Smart homes can be classified according to the types of equipment and systems installed. The major targets are improving comfort, dealing with medical rehabilitation, monitoring mobility and physiological parameters, and delivering therapy [20]. They can implement medication dispensing devices in order to ensure that necessary medications are taken at appropriate time [21].

Telehomecare technologies have been reviewed by several authors, and fall broadly into three generations: [22]

- First-generation systems are designed to reduce anxiety among elderly and high-risk patients and reduce their use of primary healthcare services. Typical technologies include personal alarm systems and emergency response telephones that make a voice connection between the patient and the response center whenever a pendant alarm button is pressed.

- Second-generation systems can generate alarms without the intervention of the patient, on the suspicion that something may be wrong. These systems can continuously monitor a large number of variables sensitive to changes in functional health status, and generate an alarm when significant changes are observed. With an intelligent decision-support system using robust algorithms, false alarms are unlikely. These second-generation systems are unobtrusive, do not require direct patient participation and can be integrated with evolving "smart home" technology for home automation, security and environmental control.
- Third-generation systems attempt to deal with issues of loneliness and quality of life of patients by creating a virtual community of clients, carers, healthcare providers and other community services, connected via the telephone, interactive television, and the Internet.

All these systems maintain a certain level of independence, thus providing a better quality of life for the resident and his close relatives. However, the recipients of smart homes are not just those with severe pathologies or chronic illness; there are also those who simply want a better quality of life. [20]

Smart homes contribute to the support of the elderly, people with chronic illness and disabled people living alone at home. This new mode of health assessment can improve the quality and variety of information transmitted to the clinician. Measures of physiological signs and behavioral patterns can be translated into accurate predictors of health risk, even at an early stage, and can be combined with alarm-triggering systems as a technical platform to initiate appropriate action [23].

Another important advantage of the smart homes is that they allow the elderly and disabled to stay at their homes instead of moving to a costly health care facility (Figure 1). The transition to a health care facility can cause a lot of anxiety and home automation can either prevent or delay this anxiety [22].

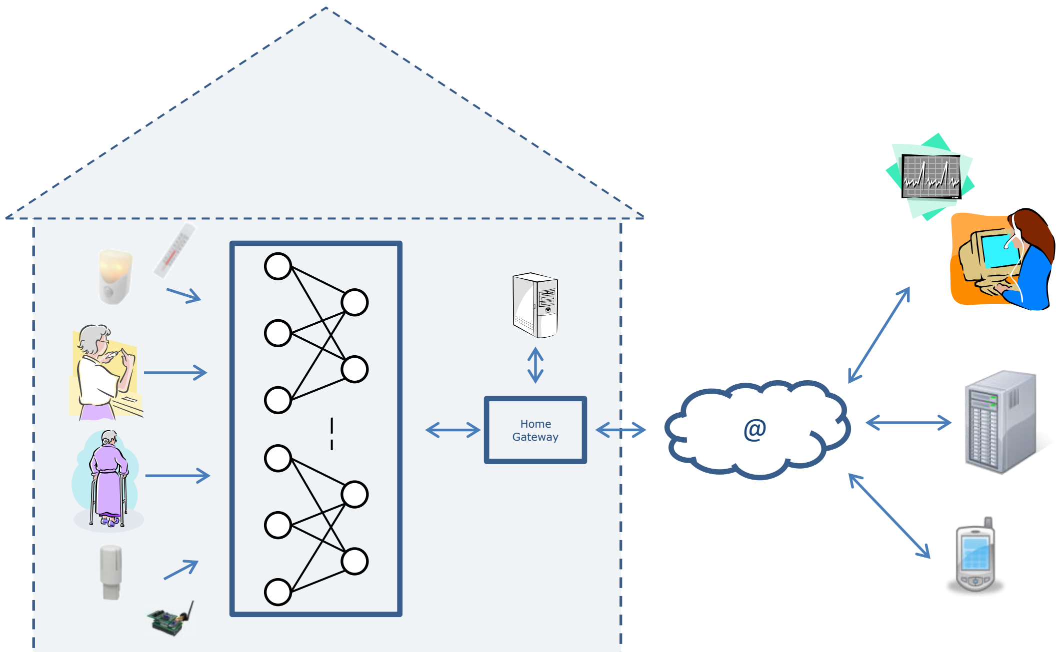


Figure 1. Healthcare home

Home automation sensor and actuators are the main building blocks in smart homes. The common home automation devices used in the smart home are lighting sensors, motion sensors, video cameras, gas and fire detectors, automated timers, automated doors and windows, chair and bed sensors, and also some more sophisticated systems, like remote keyless entry systems.

Home automation systems may be used not only to provide security, entertainment, and energy conservation, but also to make it possible for the elderly and disabled to stay at their home, by providing an activity monitoring services. Home automation systems may include automatic reminder systems for the elderly [21].

A telehomecare application is a complex system that requires the integration of various sub-systems and applications. As we said, nowadays, many telehomecare and e-Home applications are available; however, it is difficult to find the system that matches exactly the end-user requirements.

In [24] the authors presented a teleconsultation system. This system is based on web technologies and the user accesses the system over the Internet or an intranet. The system therefore is designed to do a teleconsultation, and it is not integrated with the other healthcare modules or an automatic analysis of patient's health situation. TOPCARE [25] is a telehomecare system for mobile care and homecare of high risk patients, elderly and care needing people. The system is an open telecare platform solutions for telemedicine services. The overall objective of TOPCARE is to develop technical devices and telecommunication structures and to lay the organizational groundwork for bringing co-operative health care services into the home of patients. The system comprises a telematic homecare platform (THP) backbone, the development of telematic home stations (THS) and health professional stations (HPS) and a communication server that will manage the network administration, the health professional registration at the THS, the device communication, and the Internet access.

The Interactive and cooperative telemonitoring of the dialyzing at home (DIATELIC) is a system of telemonitoring of dialyzing at home by the technology of DPCA (Dialyze Peritoneal Continue Ambulant) [26]. It was developed by LORIA in cooperation with the doctors of the ALTIR (Lorraine Association of Treatment of Renal Insufficiency) and a doctor advice of the Health insurance. Research works drove to the Diatelic system. This system is only designed especially for the dialyze patients.

The AFIRM team of the laboratory TIMC-IMAG in Grenoble developed a project called System of Information and Communication of the Intelligent Home for Health (SIC - HITCH) [27]. The objective is to monitor the patient with sensors installed at his home, by triggering on alarms in the appropriate urgent centers, in case of feelings of faintness, in falls or in abnormal situations. This system is an experimentation and simulation platform and still far from practical use in actual application.

MobiHealth [28] is an European Union project that aims to provide continuous monitoring of patients outside the hospital environment. MobiHealth targets to improve the quality of life of patients by enabling new add-value services in the areas of disease prevention, disease diagnosis, remote assistance, clinical research, physical state monitoring and even clinical research.

CodeBlue [29] is a wireless infrastructure intended to provide common protocol and software framework in a disaster response scenario, allowing wireless monitoring and tracking of patients and first responders. The system integrates low power wireless wearable vital sign sensors, handheld computers and location tracking tags. Protocols for resource naming and discovery, publish/subscribe multi-hop routing, authentication and encryption provisions are provided. It also offers services for credential establishment and handoff, location tracking, and in-network filtering and aggregation of sensor-produced data. A simple query interface allows emergency medical technicians to request data from groups of patients. CodeBlue is designed to scale across a wide range of network densities, and operates on a range of wireless devices, from resource-constrained motes to more powerful PDA and PC-class systems.

This analysis of the currently available solutions and ongoing projects motivated us to design a new telehomecare system capable of health and activity monitoring, teleconsultation, multimedia and home automation for comfort. This should benefit from the latest advances on ICT and particularly on WSN and other wireless technologies for building non-intrusive, modular, adaptive and cost effective telehomecare systems. This chapter presents the system architecture which is usable on health monitoring systems such as elderly telehomecare, and activity monitoring systems, and is capable of performing useful information about activity level of the persons. This architecture uses home automation sensors and other environmental sensors like bed and chair sensors to monitor the activity level of the elderly.

2.2 System architecture

To reach these advantages, we designed a patient oriented telehomecare system. By a user requirement analysis, we propose a Multi-tiered architecture (see Figure 2).

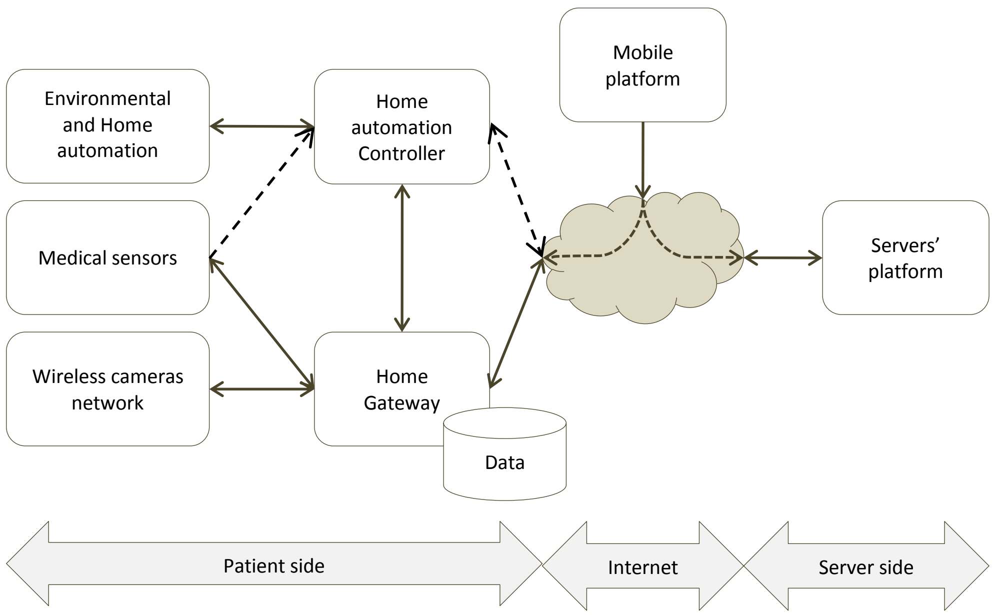


Figure 2. System's general architecture

The elements of this architecture are:

- Medical Sensors network (and/or Body Sensor Network (BSN)): This part consists of the medical instrumentation important to patients medical monitoring. This network consists of very small portable devices equipped with a variety of sensors for biological monitoring, patient localization and identification.
- Environmental Sensors and Home automation sensors network: This network must include sensors unfolded in environment (rooms, halls,

WC …). These sensors can include those of the temperature, humidity, movement, acoustics, bed and chair sensors, etc.

- Wireless cameras network: this network is used for fall detection and for confirming the detection of anomalies.
- Home automation controller: to communication with home automation sensors. The information of the sensors in this gateway is used also to activity monitoring in the Home Gateway.
- Home Gateway: It is a mini PC installed in each home. It connects body sensor network and also environmental sensors like home automation sensors and actuators to the Internet. It also includes intermediate receivers, assuring an efficient data transmission. In this gateway we do some preprocessing of the data received from medical and home automation sensors, to detect urgency anomalies.
- Mobile platform: this is a PDA or a laptop computer to receive the alerts and message by a professional, or to connect to server platform and view the patient's record. This platform can be used by persons to remote control his/her home.
- Server platform: consists of the different servers like: medical record, application, videoconference and web servers.

This system uses web services to interact between the client and the server, as shown in figure 3. This system has intention to address the need to standardize the transmission of data, by using the platform independence of web services and the structural independence of XML in the development of a web service for data transmission, processing and storage for the service of telemonitoring. By using web services, the system requires minimal additional technological elements and minimal technical requirements of support.

Figure 3. XML Data exchange

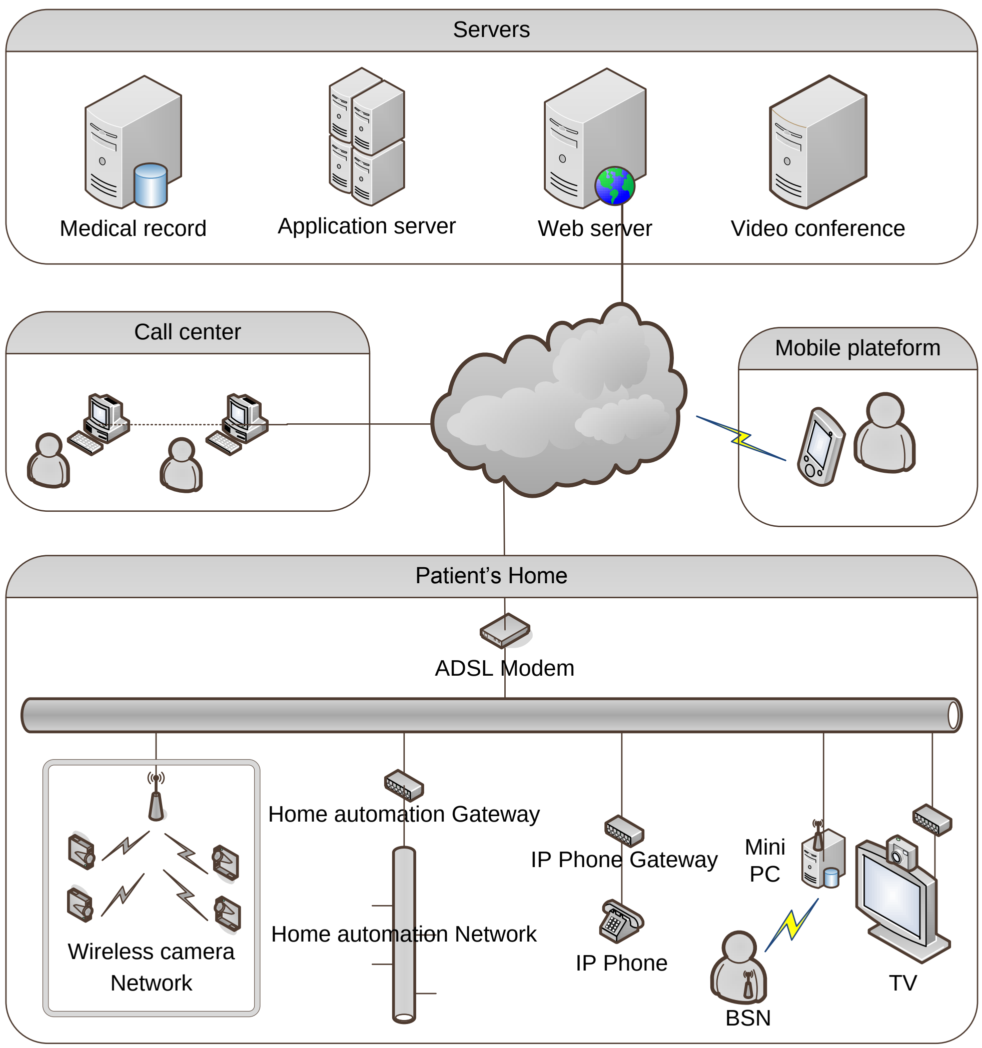


Figure 4. Different platforms

This system interfaces to healthcare providers, doctors, care-givers and the medical call centers (See Fig. 4) and also is integrated with a mobile platform for the occupant to remote control his home and another mobile platform for the doctors and nurses to view the state of their patients. Typically they will receive an alarm in case of a health problem of their patients.

As shown in figure 4, the patient record has a distributed architecture, and there is a local database in each client computer. We used a distributed patient record because in distributed database architecture the data is not stored entirely at a single physical location. Instead, it is spread across a network of computers that are geographically dispersed and connected via communications links. There is therefore a large database capacity and incremental growth, and like all distributed databases, we have a reliable, available and flexible database. The most important advantage of a distributed database for our system is that a distributed database allows faster local queries and can reduce network traffic. In this system, in each client computer we have a preprocessing of patient's data to detect the eventual emergency problems.

The different functions and features provided by the system include:

- **Storing monitored results**: as we said the system's database is a distributed database. The results received from the healthcare device are stored in the personal health database, in local PC and also in a general database in a web-server.
- **Early detection of emergency problems:** By checking the local database, if the data from healthcare devices are outside the ranges defined in their care plan, or if the data have not been received when expected, the system will trigger an alarm that may result in an automated connection with a medical call center.

- **Detection of potential problems:** The application stored in the web server, will analyze the medical record by using intelligent methods like fuzzy logic or other more complex cognitive systems, to find the potential health problem. By detecting a problem, the system will trigger an alarm that may result in an automated data transmission to a Doctor.
- **Activity monitoring:** For instance, the system allows doctors or healthcare providers to measure the physical activities of elderly. The system uses a more passive method. It senses a person's presence in a room by home automation sensors installed throughout the living areas, such as motion and presence detectors, light, magnetic and temperature sensors. Activity patterns are then analyzed in the central computer for unusual behavior.
- **Integrating with vital sign monitoring devices and sensors:** Possibility of integrating the system with the wireless devices and sensors with different communication methods such as Bluetooth, ZigBee, USB and the other technologies [30].
- **Home automation:** It gives a full control of Home automation devices, by using system's simplified interface (see chapter 5: Implementation). Alerts can also be generated by the home automation sensors, which allow an indoor localization of the beneficiary and can, thus, contribute in an anomaly detection sub-system and can help to deduct the critical situations.

2.3 Maisons Vill'Âge®

By using this system, a new concept of smart homes for the senior citizens, named in French "Maisons Vill'Âge®", is offered. Maisons Vill'Âge is a housing scheme with 25 to 50 dwelling units, organized as public or private divided co-ownership of immovable for seniors or retired person. Figure 5 shows the architecture of the system. As the figure shows, the dwelling units have the same architecture that was explained before, and they are connected to housing scheme's gateway. Each housing scheme has a control unit. The administrator of the housing scheme uses this unit to monitor the hardware and software infrastructure.

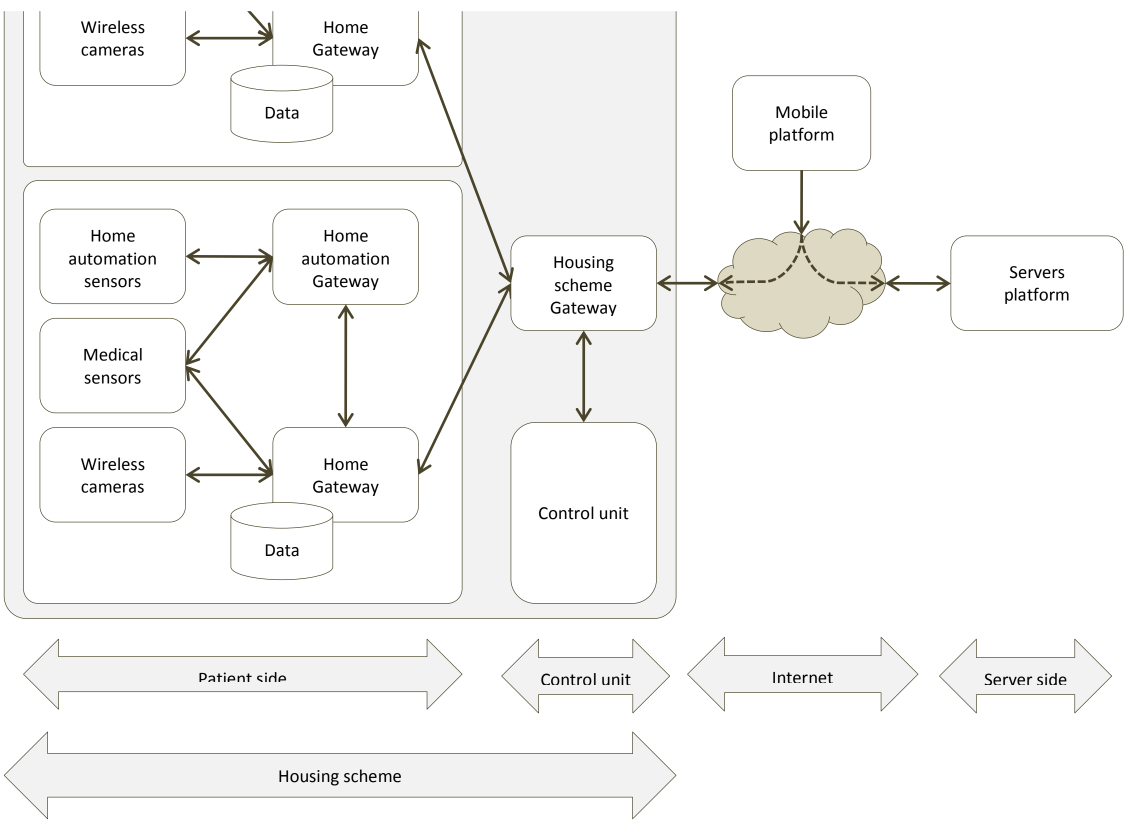


Figure 5. Maisons Vill'Âge® architecture

2.4 Discussion

In this chapter, we have proposed a distributed system to support remote medical consultations and elderly management and homecare across wide area networks and heterogeneous platforms. In this type of systems, several challenges need to be addressed.

The technological challenges regard the generic features of the wireless sensor networks, of the different supporting platforms, of the video call center, and web services, to allow a flexible and smooth interaction among those items and to anticipate further additional functionalities, interacting devices, and so on. In this respect, the methodology of specifications, development and integration of the technological features must be realized with a major concern for some important concepts, like: Interoperability, Scalability, Evolution and customization according to the users' medical and social profiles and needs.

On the other hand the design and integration effort for the technical and R&D activities of the project focus on the design of services and interfaces ensuring a maximum level of natural understanding on the way to use them, that is to say with an enhanced effort on the prerequisite and experimentation feedback's of the users who are suffering from chronic conditions and are not familiar- and sometimes reluctant- with ICT and innovation-based devices and systems. Ergonomic interface, accessibility, scalability and evolutionary solution according to the evolution of the users' needs are the important aspect of development.

Another important point is QoS requirements, like availability and response time. It is important to develop QoS specification in distributed health information systems [31]. A middleware is developed to give a dynamic and intelligent QoS to the system (see chapter 4 for more information). Due to the results of this first study, definition and revision of planned applications (scenarios improvement), behavioral intention and acceptance of designed implemented solutions and finally actual usage and acceptance evaluation should not be neglected but they are out of the scope of this PhD work and will be dealt with in future work.

Briefly, the innovation points of the proposed architecture are:

1- Fault tolerance: in the proposed architecture, there are some redundancies of the gateways in order to have a tolerance to failures. As an example, ordinary, the medical data use home gateway to communicate with the central servers, but in case of unavailability of the home gateway they can use home automation gateway to send the data.

2- Robustness and adaptability of the system to the mobility of the sensors: The system uses multi sensor data fusion to deduce the activity level of the patient.

   We performed a simulation of our system to find the advantage of the multi sensor data fusion. The simulation scenario consists of one patient and a flat of 34 m^2 with one bed room of 10 m^2, a WC of 4 m^2, a kitchen of 5 m^2 and a bath room of 3 m^2, a salon of 12 m^2. The maximum speed of the patient is 0,5 m/s. The simulation duration is 100 rounds which presents 100 days. In each round there is one abnormal situation with a random time.

   We performed the simulation in 3 steps: In the first step we use just one presence sensor in each room; in the second step, we add one bed presence sensor in the bed room, and in the final step, we add one presence sensor in the chair of the salon.

Table I shows the results, which confirm the efficiency of using the multi sensor data fusion.

TABLE I. SIMULATION RESULTS

Step	Abnormal situations	Alarms produced with the system	False alarms	Real alarms
1	100	180	149	31
2	100	135	88	47
3	100	114	50	64

Note:

For the simulation we used just static configuration of the parameters of the system, but by using an intelligent algorithm (for example using the fuzzy logic) we can reduce the false alarms.

2.5 Conclusion

As part of this thesis contribution, a telehomecare system is developed to remote monitor activity and health state through sensor networks and heterogeneous platforms. The segment of population we are targeting is the senior citizens. This system uses home automation sensors and other environmental sensors like bed and chair sensors to monitor the activity level of the elderly.

We conducted a study in total of 24 dependent elderly people experiencing several types of limitations (sensory, physical, cognitive or none) and leaving in 3 different residential home care units in France and 13 healthcare professionals (4 managers, 4 nurses and 5 caregivers) were interviewed in order to measure the user needs and technology acceptance.

This study shows that all the healthcare professionals have a positive attitude toward dedicated means for ICT practical application (50% "totally agree", 33.3% "strongly agree" and 16.6% "lightly agree"). Majority of interviewed elderly residents, in this study, declare not knowing what would be their family attitude toward ICT use (45.83%); 37.5% think that their family would be favorable to ICT use and 16.67% think that the family would be unfavorable to ICT use, but this study with the projected family attitudes of elderly people shows that the limitations experienced in everyday life do not lead to a favorable or unfavorable category of ICT user [32].

Chapter 3 FLoR

a Fuzzy logic based Adaptive Clustering and Routing

As previously mentioned, WSN is a key component that we would like to introduce in a telehomecare system. In such a network, data are transmitted directly or indirectly through other intermediate nodes. The data collection is done either periodically or event-triggered. Some data are highly vital and need guaranteed performance. In WSN, limited energy resource, failure of the nodes or the communication link and mobility of the nodes are the constraints to provide a reliable communication. This necessitates the use of algorithms that are able to consider many criteria with multi-objective optimization to adapt to topology changes of the network. Classic heuristic approaches are generally not suitable, because of the long online execution time.

This chapter presents FLoR, an adaptive clustering and routing algorithm which is based on Fuzzy Logic. Dynamic load balancing, Mobility management and high data delivery ratio are the main characteristics of FLoR. This algorithm fits especially well to networks that use sensor nodes for data collection and in which the data delivery ratio is important and the nodes are mobile, like health monitoring sensor networks.

3.1 Introduction

Wireless sensor networks have many applications. Smart homes, interactive virtual worlds are only a few examples. An example is healthcare network. These systems present important research and technical challenges. In a healthcare network, because of mobility of the patients, all the sensors attached to them are mobile; therefore the mobility management and localization are important points in these networks. Figure 6 shows a typical architecture of a healthcare system. In this system, the communication backbone consists of the mobile or fixed nodes and also the fixed or wired nodes or sensors attached to the other persons, like doctors and nurses.

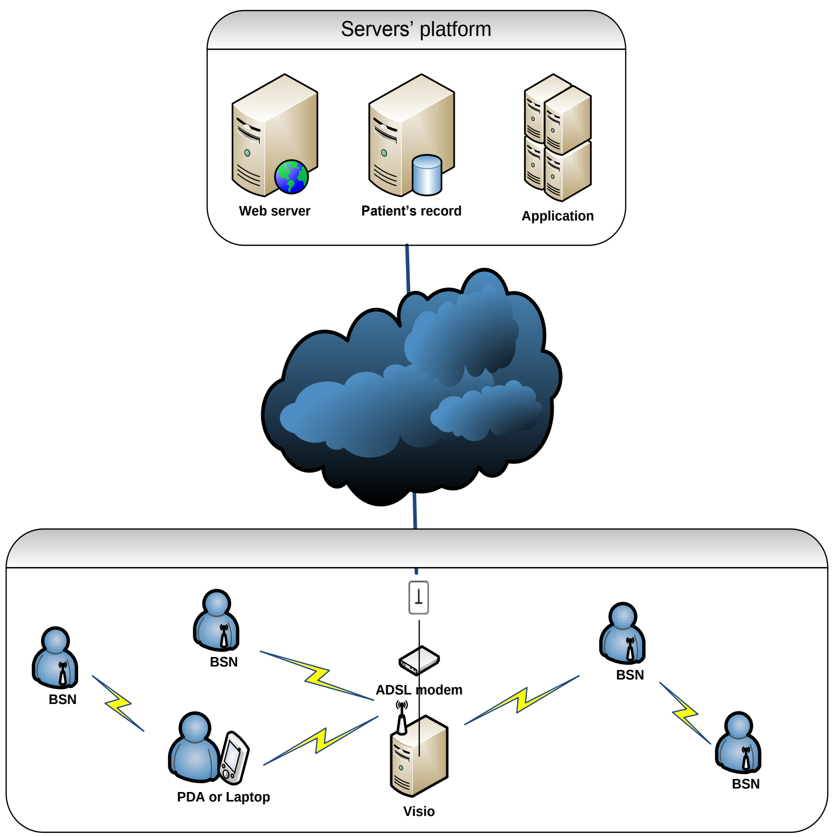


Figure 6. An example of mobile network

In fact "Maisons Vill'Âge®"(see chapter 2 for more details) is a more complex system. The first objective is to provide QoS solutions for this concept. Figure 7 presents the general system architecture of "Maisons Vill'Âge®".

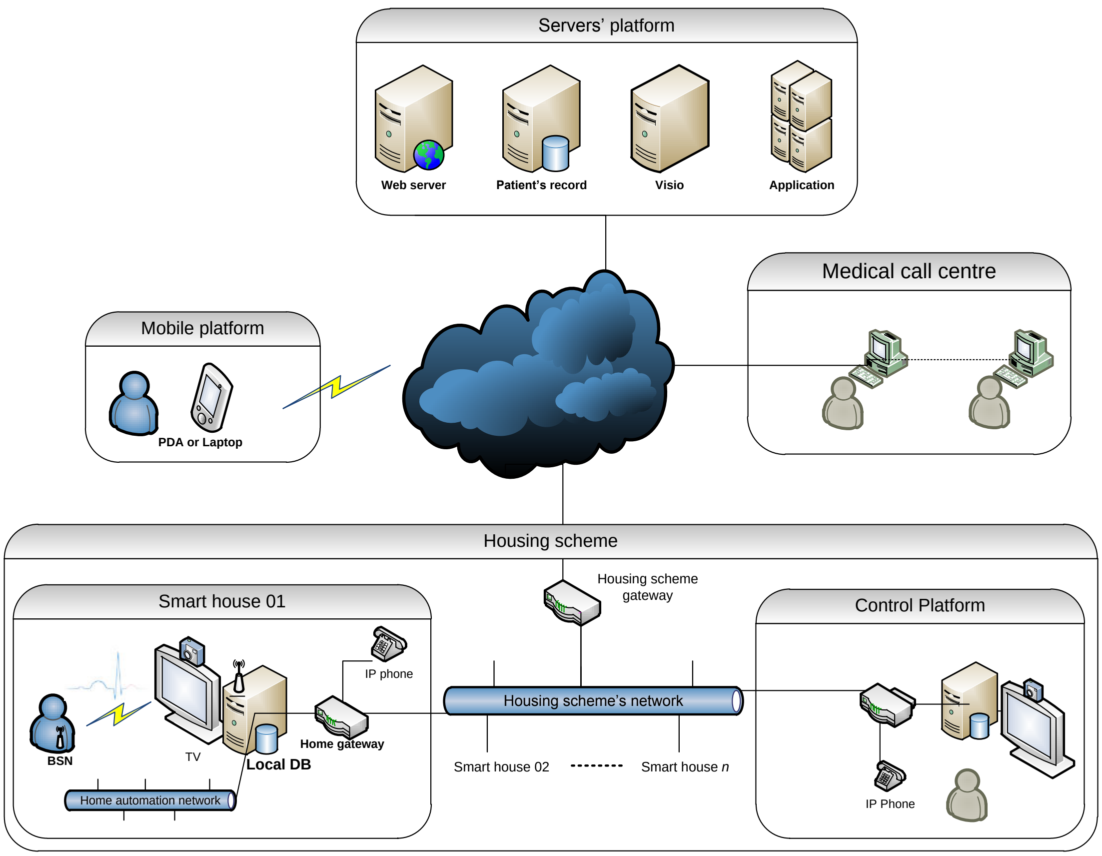


Figure 7. System architecture - "Maisons Vill'Âge®"

These systems need an adaptive routing and localization that can support mobility of the sensors and changes of network topology when patients are moving. Based on the topology of the sensor networks, the routing protocols fall into two classes: (1) flat routing protocols and (2) hierarchical routing protocols.

In flat networks, each node typically plays the same role and sensor nodes collaborate together to perform the sensing task. Due to the large number of such nodes, it is not feasible to assign a global identifier to each node. This consideration has led to data centric routing, where the BS (base station) sends queries to certain regions and waits for data from the sensors located in the selected regions. Since data is being requested through queries, attribute-based naming is necessary to specify the properties of data [33].

Hierarchical or cluster-based networks are well-known techniques with special advantages related to scalability and efficient communication. As such, the concept of hierarchical architecture is also utilized to perform energy-efficient routing in WSNs. In a hierarchical architecture, higher energy nodes can be used to process and send the information while low energy nodes can be used to perform the sensing in the proximity of the target. This means that creation of clusters and assigning special tasks to cluster heads can greatly contribute to overall system scalability, lifetime, and energy efficiency. Hierarchical routing is an efficient way to lower energy consumption within a cluster and by performing data aggregation and fusion in order to decrease the number of transmitted messages to the base station [33].

Forming dynamically the optimized clusters and adaptive routing by choosing the best route are the main problems to solve. The next section discusses the challenges in detail.

3.2 Challenges and objectives

Due to the advantages of cluster-based networks, we chose that as network architecture for "Maisons Vill'Âge®". Figure 8 shows the "Maisons Vill'Âge®" cluster based network architecture.

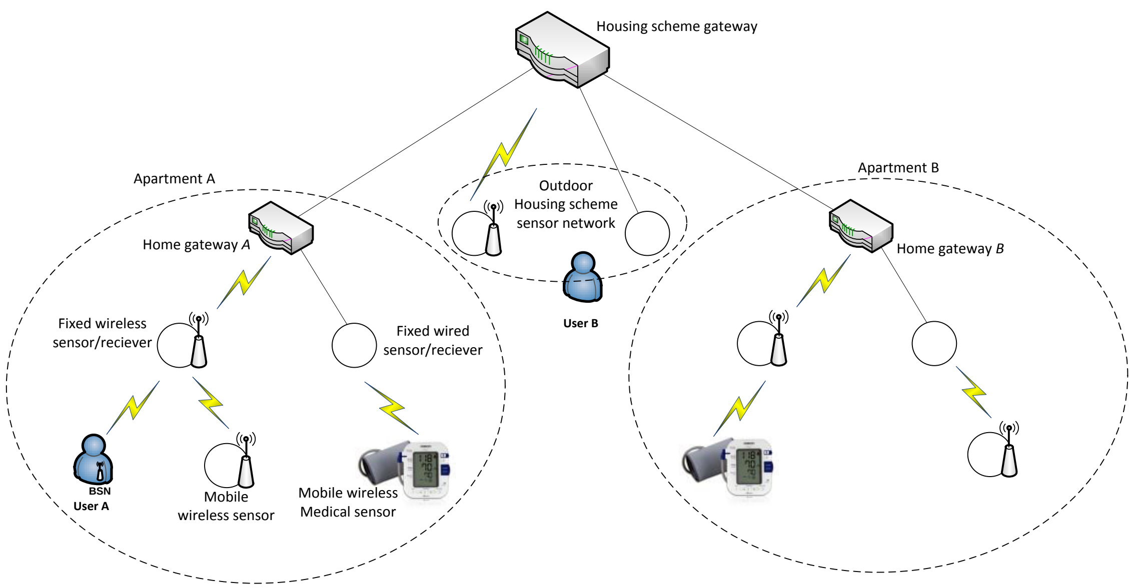


Figure 8. System architecture - "Maisons Vill'Âge®"

As the figure shows, in this network, users and medical sensors are mobile and the fixed environmental sensors can be wireless or wired. As the user leaves his apartment, the system must guarantee the continuous localization and health monitoring in whole housing scheme. For this purpose, there is an outdoor housing scheme sensor network which consists of wired or wireless sensor nodes.

There is a significant gap between the current sensor network technologies and the special needs of telehomecare systems. Generally the system must be able to provide QoS requirements by considering the technical challenges in WSNs, in order to provide a robust dynamic clustering and routing (see figure 9).

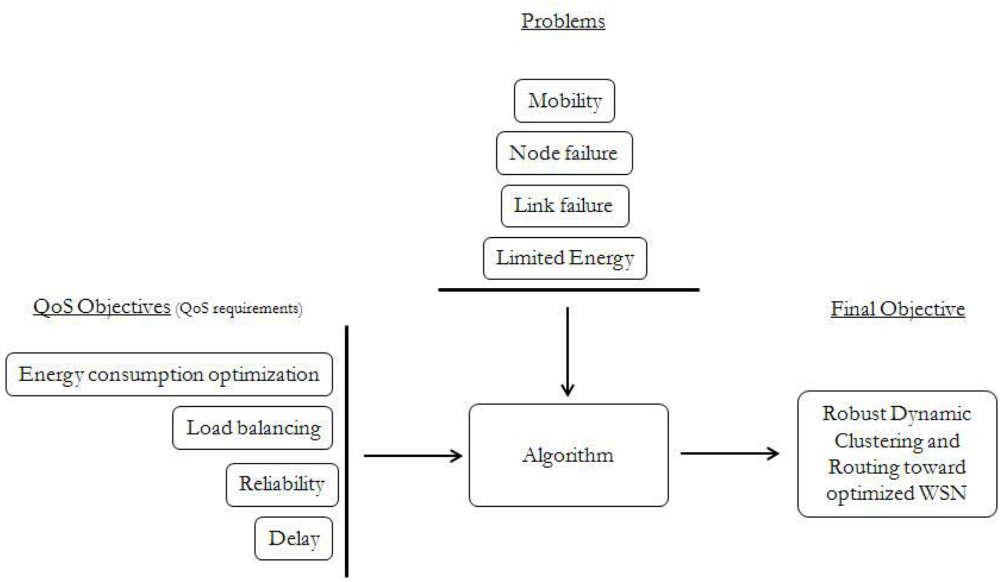


Figure 9. Problem description

The final objective here is providing a robust dynamic clustering and routing algorithm. In this respect and regarding to our objectives, providing QoS requirements must be considered. QoS requirements differ from one application to another. In our project, they can be specified in terms of energy consumption optimization, load balancing, minimizing the delay and providing the reliability.

One of limits of wireless sensor nodes is their inherent limited energy resource. The load balancing averages the energy consumption and this extends the lifetime of the whole sensor network by extending the time until the first node is out of energy. Load balancing is also useful for reducing maintenance time and message exchange traffic. In a cluster-based network, the lake of the load balancing method can cause some clusters with high node density and other clusters with low node density. It is clear that in a cluster with a high node density, energy consumption in cluster head will be very high and it will die rapidly, this can cause network partitioning.

Despite of many research works, failure management in sensor networks is still an open issue. Generally, the communication in sensor networks, with battery powered nodes, suffers from failure, low transmission power even more so than in general wireless networks. In addition, network dynamics resulted by node mobility and node state transitions due to the use of power management or energy efficient schemes (e.g. Sleep mode) may be wrongly detected as node or wireless link failure.

In healthcare networks, failure of a sensor can be wrongly detected as a health anomaly. These two cases can cause false alarms. Any failure in a healthcare system can lead patient situation to a critical level. This failure may occur in a sensor, computer, and router or in the communication link. The traditional failure management systems fail to address these challenges because they are simply made for wired local area networks and limited-scale systems, where the mobility and energy management schema do not exist. In particular, traditional implementations are not designed to cope with a large number of processes, a high message loss rate, a dynamic network topology, and the unpredictability of wireless sensor networks. Message loss, Message saving, However, the traditional failure detectors and management systems assume that all of the nodes of the network are synonymous, that means there is no difference between a node that was failed *n* times in *t* hours, with a node that was failed *m* times (m>n) in the same period of time. That will causes an unclear decision making and electing between different nodes. In addition, in the traditional failure detectors, when a node failed, it will be assumed as a dead node and we don't have a return of the node. That will be a restriction, for example, when a node is in maintenance.

3.3 Toward a multi-objective and multi criteria approach

As mentioned before, the main objective is proposing a system for forming dynamically the optimized clusters and adaptive routing.

Several dynamic clustering strategies have been proposed in the literature. ZRP [34], LEACH [35] and HEED [36] are some examples of clustering protocols. In [34], the zone routing protocol (ZRP) a hybrid strategy, is proposed by Haas and Pearlman which attempts to balance the trade-off between proactive and reactive routing. The objective of ZRP is to maintain proactive routing within a zone and to use a query–response mechanism to achieve inter-zone routing. In ZRP, each node maintains its own hop-count constrained routing zone; consequently, zones do not reflect a quantitative measure of stability, and the zone topology overlaps arbitrarily.

LEACH [35] is an application-specific data dissemination protocol that uses clustering to prolong the network lifetime. LEACH clustering terminates in a constant number of iterations (like HEED [36]), but it does not guarantee good cluster head distribution and assumes uniform energy consumption for cluster heads. In contrast, HEED makes no assumptions on energy consumption and selects well distributed cluster heads but HEED assumes quasi-stationary nodes.

In brief, the existing clustering protocols use many assumptions which make them not able to address the needs of telehomecare application: Some algorithms assume a network with fixed or low mobility nodes, which is not the case in a telehealthcare network; Some algorithms are based on centralized control that makes them not scalable; Some algorithms use periodic rounds to change cluster head and elect a new one. The new cluster head will be fixed for one round, but in an ad-hoc network with dynamic topology, cluster topology may change during this period, and in this case a new cluster head must be elected. Therefore this type of algorithms will be good for networks with fixed or very low mobility nodes, but not for telehomecare networks.

To balance the load in the network, most of the clustering protocols use different parameters to choose cluster-heads. Cluster ID [37], connectivity degree [38, 39] and periodical cluster heads election [35] are used in order to share the load among all the nodes of the network. By applying cluster ID or highest connectivity methods, the same node may be chosen as cluster-head every time, and that will result in draining its energy very fast.

Regarding the routing, the existing algorithms can be classified into static (non-adaptive) and dynamic (adaptive) algorithms. Static algorithms do not make routing decisions based on current state and topology of the networks, in contrast, dynamic algorithms change their routing strategy either when the state and the topology of the network changed or periodically. These adaptive routing algorithms are more applicable in the mobile wireless sensor networks because of their dynamic nature.

Numerous parameters have to be considered when choosing a route in sensor networks. The route choice can be characterized by various criteria (See figure 9). Existing algorithms decide the routing path by using mathematical crisp functions [40, 41] which take into account just the crisp and precise parameters (like distance, load, etc.). But real life data are not always crisp, and all descriptions cannot be always expressed or measured precisely. These algorithms also do not take into account the essential criteria of optimal routing like unavailability rate of the node.

One possible solution for this multicriteria problem is using Fuzzy Logic [92]. Fuzzy systems usually describe complex technical systems in a simple way tolerating impreciseness. Linguistic rules are used to describe the model. It is not necessary to know the exact mathematical function which describes the behavior of the system, because fuzzy logic has potential for dealing imprecision in data using human reasoning without needing complex mathematical modeling. Fuzzy logic on the other hand is capable to combine many criteria to decide a route. The values of the real world parameter are mapped to fuzzy values by fuzzification and then processed by the interference machine. These fuzzy results are transformed into crisp values again by defuzzification.

The discussed objectives and challenges and the lake of a satisfying approach for our requirements [33], motivated us to propose FLoR (Fuzzy Logic based Adaptive Clustering and Routing for Wireless Sensor Networks), a new cluster-based routing which can manage the mobility and unavailability (caused by failure or mobility) of the nodes. The algorithm uses a multicriteria decision to form the clusters and manage the mobility and unavailability of the nodes and because of this multicriteria decision strategy it uses fuzzy logic to facilitate the decision process. To make it scalable the protocol is totally distributed, it has also a load balancing part. All event-processing and decision-making processes of FLoR use fuzzy logic.

3.4 Fuzzy Logic and its applications in the networks

As discussed, in our proposal, we use fuzzy logic because it is capable of making real time decisions with incomplete information, even in real time. Moreover fuzzy logic can be used for context by blending different parameters – rules combined together to produce the suitable result. In this section we review some examples of applications of fuzzy logic in the clustering, routing and QoS. For reading comfort, we also give in appendix A a short recall of the basic notions in Fuzzy logic.

3.4.1 Clustering

In [42], a fuzzy logic approach to cluster head (CH) election is proposed based on three descriptors - energy, concentration and centrality. In this approach the cluster-heads are elected by the base station in each round by calculating the chance each node has to become the cluster-head by considering three fuzzy descriptors – node concentration, energy level in each node and its centrality with respect to the entire cluster. This technique is proposed to use in LEACH [35], but it cannot support the mobility of the node and in addition it is centralized algorithm and therefore it cannot be scalable.

In addition, in a distributed cluster based approach each node chooses its parent by processing its local information therefore centrality of a node cannot be suitable to its decision, this parameter will be good for a centralized algorithm to reduce the routing hops between CH and nodes in the same cluster, but a node cannot have a general view of network and thus cannot evaluate this parameter for a CH. Additionally, in a dynamic network with mobile nodes the number of neighbors is not a good descriptor, because location of the node is not fixed and we cannot be sure about number of neighbors at each moment.

This approach uses a central computing of the cluster-head election but as we know a system with a central control, has just one critical point and also it has the problem of scalability. In addition, the nodes are obliged to send the information to the base station, in order to update the information like their *energy* and their location also if the nodes are mobile, and this demand energy consumption and a high traffic of data transmission between the nodes and the base station, in dynamic networks. To reduce this problem, the authors have assumed that the nodes have minimal mobility thus sending the location information during the initial setup phase is sufficient. Therefore this approach is not suitable for large scale dynamic and mobile sensor networks.

Another fuzzy based clustering protocol for distributed wireless sensor networks is proposed in [43]. The protocol is called Fuzzy Self Clustering Algorithm (FSCA) and is based on emergent protocol called An Emergent Algorithm for Highly Uniform Cluster Formation (ACE) [44]. FSCA is designed to increase the lifetime of the sensor networks. To reach this objective, the protocol tries to distribute the clusters uniformly by re-clustering the whole network repeatedly to provide the clusters without any overlap between them and also to reduce the number of nodes which are not member of a cluster.

The algorithm classifies nodes into three types: *Cluster-Head*, *Cluster-node* (member of a cluster) and *Unclustered* (that is not a member of any cluster). Each sensor node uses the *energy level*, *local density* within its sensing range and *time* as parameters for clustering, re-clustering, and merging existing clusters by using fuzzy logic. (The re-clustering process is similar to the clustering process except in clustering process all nodes start in *Unclustered* state, but in the re-clustering process the node starts the algorithm with its current state). To optimize the power consumption, the authors have found that re-clustering the whole network every 30 sensing rounds increases the network lifetime, the paper doesn't give more information about this number, but in the large scale and dynamic networks, re-clustering the whole network will add overload to the network and will demand energy consumption to send the messages and also to compute and election process. Finally, the protocol doesn't work in mobile networks and is designed for networks with fixed sensors. The authors have assumed also that the sensor nodes are homogeneous and they are unaware of their locations.

3.4.2 Routing

[45] presents a routing analysis algorithm for wireless sensor networks which uses a fuzzy logic system at each node. The fuzzy system is used to determine the capability of the node to transfer data. The objective of the algorithm is to automatically select the data dissemination protocol that better meets application-specific requirements while minimizing the network resource consumption.

In order to maximize the lifetime of the sensor networks and to help for the selection of node to forward packets to the destination, the fuzzy system utilizes four fuzzy variables: relative energy levels, distance of the node from source node, the distance of the node from shortest path and traffic load. The two fuzzy variables, distance and distance from the shortest path lead the algorithm to use GPS (increase the cost) or a Localization technique. As the variables must be periodically updated in order to be able to make clear decisions, using GPS or localization technique can increase the energy consumption in a dynamic network.

In the simulation of the algorithm these problems have not been considered by the authors.

3.4.3 QoS

FuzzyMARS, a fuzzy logic based semi-stateless QoS approach is presented in [46]. The main objective FuzzyMARS is to guaranty a service differentiation delivery with low delay transmission. FuzzyMARS is composed by a best-effort traffic regulation system which is based on fuzzy logic and an admission control mechanism for real-time traffic control.

FuzzyMARS uses 2 linguistic variables: 1) The delay-measurement obtained as feedback from the MAC layer, as input and 2) traffic regulation rate as output. Using the feedback from MAC layer is very sensitive to node or link failure, node mobility and topology changes of the network. If any failure occurs in the network, the information received by the source node, may be false and this will cause an incorrect decision to reject or accept the QoS requirements. The system can be used in the networks by low mobility nodes.

FLC-QM (Fuzzy Logic Control based QoS Management), another fuzzy logic based QoS management method is proposed in [47]. FLC-QM takes advantage of the feedback control technology to deal with the impact of unpredictable changes in traffic load on the QoS of WSANs.

This approach uses a fuzzy logic controller which dynamically adjusts the sampling period of relevant sensor in order to keep the deadline miss ratio at a desired level. Inside each source sensor node, the FLC-QM utilizes a fuzzy logic controller to adapt sampling period to the deadline miss ratio which is associated with data transmission from the sensor to the actuator. In this approach failure and mobility of the nodes was not considered.

3.5 FLoR – Fuzzy Logic based Adaptive Clustering and Routing for Wireless Sensor Networks

This section presents FLoR, our adaptive clustering and routing algorithm. As a result of our discussions in the last sections, Mobility and Failure management and Load balancing are main important problems to be addressed in ad-hoc sensor networks. To address these problems, FLoR has 5 main parts: Fuzzy logic decision making, Clustering (Cluster-head election), Mobility management, Load balancing and Unavailability management. Figure 10 shows the main parts of the protocol. As we see in this figure, fuzzy decision making is the basic part of our proposal. That means, the 4 other parts of the protocol, use fuzzy logic to make decision or to process an event.

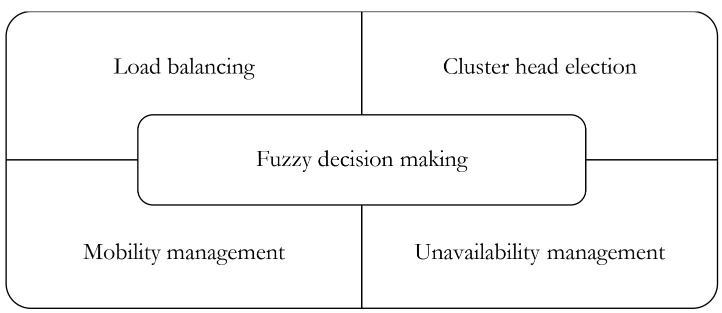


Figure 10. Building blocks of the protocol

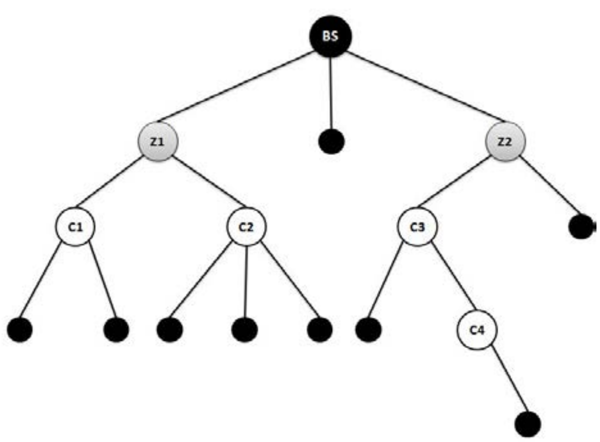


Figure 11. Zone and Cluster example

3.5.1 Definitions

Before to explain our proposal, in this section we present some definition that we use in our protocol:

- {ALL}: Set of all sensors of the network.
- {Unknown}: the set of the nodes with unknown situation.
- {myNeighbours}: Neighbours of a node.
- {myChild}: Child nodes of a parent node.
- A→B: Direct communication between A and B
- Base-Station (BS): Central computer that monitor the network or gather the information generated by sensors. (BS in Figure 11)
- Zone-Head (ZH): Node (mobile or stationary) that communicates directly with BS (Z1 and Z2 in Figure11):

  $\forall\, n \in \{ALL\} \mid n \rightarrow BS \Rightarrow n$ is a ZH

- Cluster-Head (CH): Node (mobile or stationary) that can communicate with one or more ZH or a node with some children that can communicate with other CH(C1,C2,C3 and C4 in Figure 11):

  $\forall\, n \in \{ALL\}, \exists\, m, z \in \{ALL\} \mid n \rightarrow m \land z \rightarrow n \Rightarrow n$ is a CH

- Zone: set of one or more cluster. Each ZH constructs a Zone.
- Leaf-Node (LN): a node without child (Black nodes in Figure 11):

  $\forall\, n \in \{ALL\}, \nexists\, m \in \{ALL\} - n \mid m \rightarrow n \Rightarrow n$ is a LN

- {Unknown}: the set of the nodes with unknown situation.
- Level: Level of a node is the number of hops between the node and BS.

- Mobility: This is a parameter to evaluate the movement of a node. The complete description of this parameter can be found in the next sections.
- Quality of Link (QoL): this parameter shows the Reliability of the connection between a node and its parent.
- Successful clustering: In this algorithm, we have a successful clustering if and only if: $\forall\ n \in \{ALL\}, \exists\ m \in \{ALL\} - n \mid n \rightarrow m$

  and also: $\forall\ n,m \in \{ALL\} : (n \rightarrow m) \Rightarrow m \in \{ZH\} \cup \{CH\}$

- Invite: it is a message, between the nodes to exchange the information. This message is used by a ZH or CH to invite the other nodes to join them. It will be used by nodes to find a new parent also.
- Find: is used by a node to find a parent.
- Hello: is used by a node to announce a change or event to its neighbors.
- Join: this message is used by a node to answer to an Invite message. If the node has just one possible candidate to choose as its parent, it will indicate that, in this message.
- Quit: a node sends this message to its parent node to inform leaving it.
- Join_Other: It is a message that a ZH or a CH sends to one of its children to ask him to find another parent. This will occur when the ZH or CH received a new request of join from a node with no other possible parent, and the admission condition is not satisfied. Therefore it must reduce its load by reducing number of its children.

3.5.2 Mobility management and new definition for mobility

In an ad-hoc sensor network with mobile nodes, we must be able to detect the movement of the nodes in order to have a correct image of network at any time. We know that the nodes, in these networks, are not integrated with systems like GPS, but without GPS how can we detect the movement of a node? Assume that distance between two nodes A and B in time t is X, and in time t+1 the distance is Y where Y>X. By using this information we cannot say which one moved? A, B or both of them? Without a lot of calculations or a GPS system, we cannot answer this question.

In order to solve this problem we proposed a new parameter named Mobility. This parameter shows frequency of parent, level or zone change of a node. (Number of CH or level change of a node in his life time). Therefore each time that the node changes its CH or his level, it must increment value of a variable named "Change" and divide it to his lifetime to find the Mobility. It is clear that the mobility of a fixed node can be greater than zero, because of the mobility of its parent.

3.5.3 Load Balancing and our definition of load

Our load balancing strategy considers the cumulative load of data traffic from child nodes in a load tree on their parent nodes. We use *Load tree* and *admission condition* for load balancing.

3.5.3.1 Load tree

Figure 12 shows a sample *load tree*. The *load tree* is rooted in the base station. The load of child sensor nodes adds to the load of each upstream parent in the tree. Hence, the sensor nodes nearest the base station will be the most heavily loaded. The goal of this load balancing technique is to evenly distribute packet traffic generated by sensor nodes across the different branches of the Load tree. But here Load has a special definition.

Load is the average of the QoL (see next section) between a node and his children. It is a new definition that can be used as a new parameter in QoS. In a load tree, the weight of each link, in load tree is QoL between each node and his parent, and the load of each node is the sum of the QoLs between the node and his child. For a node n, we have:

$$n.Laod = \frac{(\sum_{n.Child} n.Child.Load) + (\sum_{n.Link} n.Link.Value)}{\text{n. numberLowLevelCH} + \text{n. numberLowLevelLN}} \qquad (1)$$

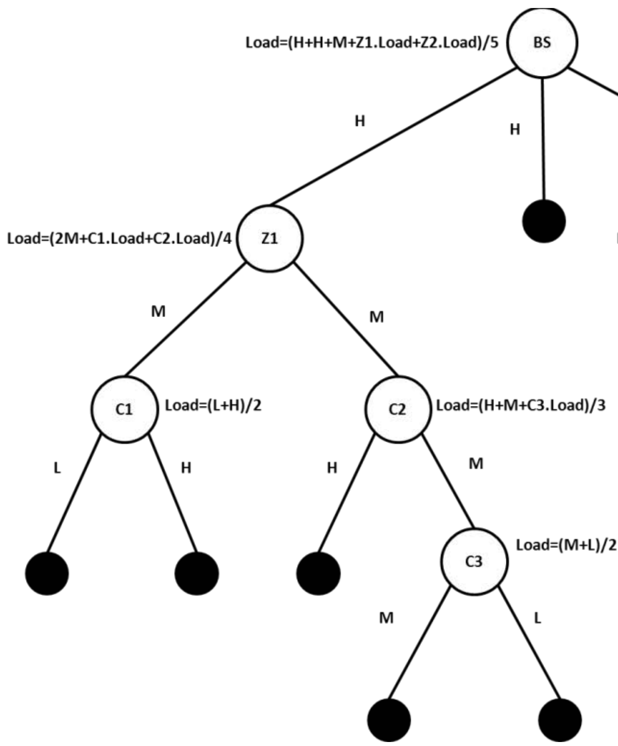


Figure 12. A load tree

As an example, in figure 12, the node C1 has 2 children. The QoL betwwen these nodes and C1 are L and H. therefore the Load of C1 is (L+H)/2

3.5.3.2 Admission condition

In order to balance the load between nodes of the network, we use admission condition, which is the condition of accepting a new child node in node n:

$$(\text{n.Load} + \text{NewChild.QoL}) / \text{n.numberLowLevelLN} + 1 \leq \text{n.QoL} \qquad (2)$$

In this formula the number of low level nodes means the number of all the nodes that are connected to this node directly or indirectly.

3.5.4 Unavailability management

As the unavailability of a node may have many reasons like mobility of the node and node or link failure, it is clear that in a real environment, we cannot control or reduce the number of unavailability of the nodes in a network but it's possible to manage it. Implementing failure detectors over local networks is, by now, a rather well-known issue, but it is still far from being a solved problem with wireless sensor networks. But traditional solutions fail to solve important problems such as the potentially large number of monitored processes, the higher probability of message loss, the ever-changing topology of the system, and the high unpredictability of message delays [48].

A number of high level applications for wireless sensor networks have been envisioned [49-51]. Fault tolerance in measurements by a group of sensors, was first studied by Marzullo [52]. Marzullo proposed a flexible control process program that tolerates individual sensor failures. Issues addressed include modifying specifications in order to accommodate uncertainty in sensor values and averaging sensor values in a fault-tolerant way. [53] developed an algorithm that guarantees reliable and fairly accurate output from a number of different types of sensors when at most k out of n sensors are faulty. The results of the scheme are applicable only to certain individual sensor faults and traditional networks. They are not extendable to the reliability needs in complex network levels and most importantly; they do not address the reliability issues that are induced by the ad-hoc nature of the wireless sensor networks.

For these reasons, we use a technique to improve our routing algorithm by considering a history of nodes' unavailability. By using this technique, our routing algorithm dynamically adapt to nodes' unavailability. To do this, we use *Unavailability* parameter which presents the unavailability rate of a node detected by its neighbor. This parameter will be computed by using the number of node's unavailability during its lifetime. It is also a fuzzy variable that has 3 levels: High, Medium and Low. Each time that the node detects unavailability in a neighbor, it updates *Unavailability* parameter of this node:

$$Unavailability = fuzzy\ (n\ /\ L) \qquad (3)$$

where *fuzzy* is a function to convert decimal value to fuzzy value, *n* is number of neighbor's *Unavailability* and *L* is our life time, therefore the *Unavailability* parameter has different value for each node in the other nodes. In each network, due to the mobility or *Unavailability* rate of the nodes, Base Station will defines an update period, in which each node will update the *Unavailability*, therefore *this* parameter is really dynamic parameter that can change not only from *Low* to *High* but also from *High* to *Low*.

3.5.5 Cluster head election

We use four parameters: *Energy level* of the node (Battery charge), *Mobility*, *Quality of Link* - *QoL* (R*eliability* between a node and his parent) and the *failure,* to evaluate a node that is candidate to be a ZH or CH. These parameters will be the *Fuzzy Logic Descriptors* and each of them has three possible values: *low, medium, high*. Therefore we have maximum 81 rules to evaluate a node. The result of the rules will be R*eliability* with 3 possible levels: *Low, Medium, High* (Figure 13).

In each *Invite* message the node will send necessary information to be evaluated by the other nodes, as like as: *Energy level* and *QoL*, and the node will compute the QoL of the connection between candidate and itself. The QoL of a node is

Reliability parameter that he was calculated for his parent. This parameter helps us to choose the best parent node, a node with maximum energy, maximum stability, and higher reliability of connection. By finding the Reliability of a candidate we must evaluate the chance of the candidate to be a parent. To restrict depth of network's tree when a node receives more than one Advertisement, it will choose the node with smaller level, therefore we use:

$$Chance = Reliability \,/\, Level \qquad (4)$$

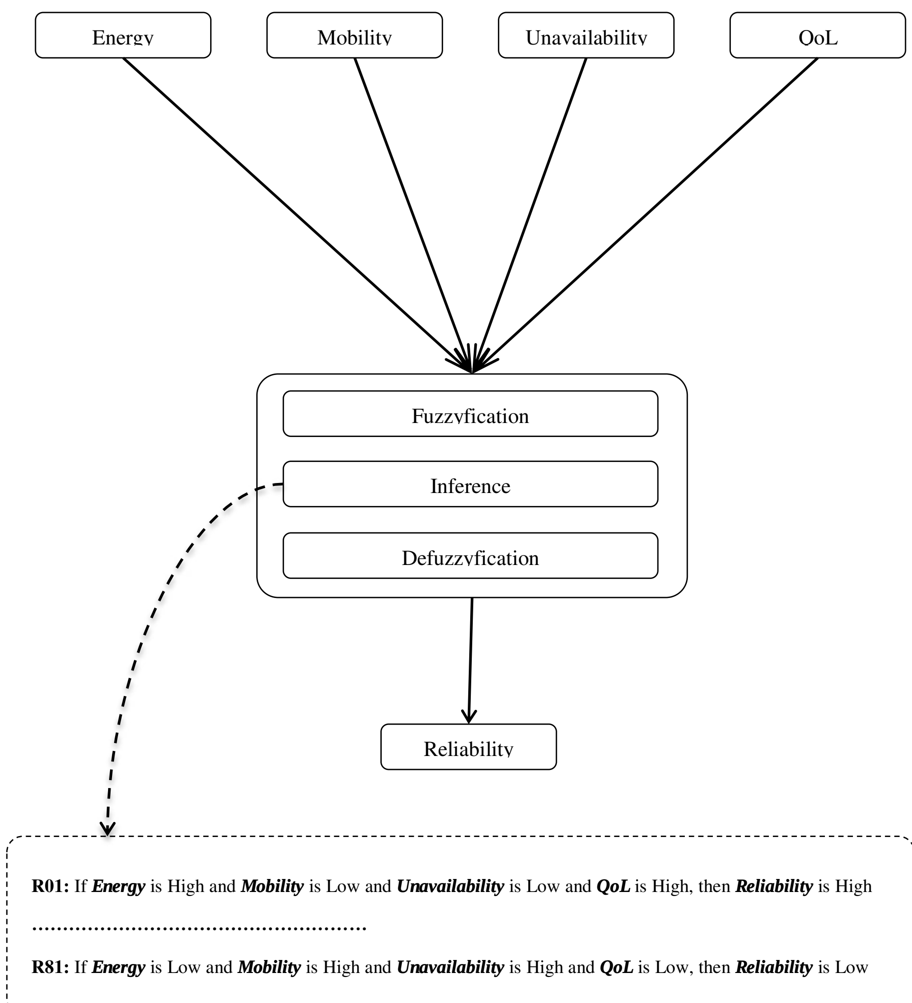


Figure 13. Cluster head election

To see the detail of fuzzy computation, we take a simple example. Node *n* received an *Invite* message from *CH* and it evaluates CH by CH's *Unavailability* and *Mobility*. Let's *CH.Unavailability*=45 and *CH.Mobility*=55.

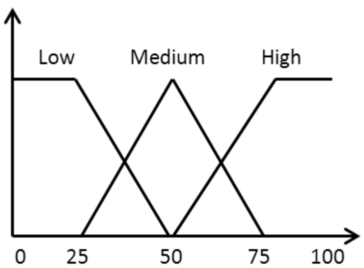


Figure 14. Fuzzy set of the example

Figure 14 shows the fuzzy set used in the example. For each fuzzy variable, there are three possible values: Low, Medium and High.

Note:

To simplify the example the break points of the fuzzy set are chose 0, 25, 50, 70 and 100, but choosing these points is a part of configuration of each system. That means for each system, we must find the best values. Obviously, the values of these points have a direct relationship with the performance of the system. Finding the optimal values of these points is one of our future works. We will investigate to propose a system with a learning period in order to find the points dynamically.

Figure 15 shows the different steps of the process. These processes are:

1- Fuzzification: By using the fuzzy sets (figure 14) and the fuzzy rules, system finds the membership degree of values to each fuzzy set. As shows the figure, membership degree of Unavailability is 0,4 to the Low fuzzy set and the Mobility has a degree of membership of 0,7 for Medium set and 0,2 for High fuzzy set.

2- Apply fuzzy operators: Here in our example, the 2 rules have AND operator which means minimum. Therefore, for each rule we find the minimum value (For Rule 1 the minimum value is 0,4 and for the second rule, 0,2) and find the area of the fuzzy set of output of fuzzy rule (For Rule 1 the output fuzzy set is Medium and for the second rule the output is Low).

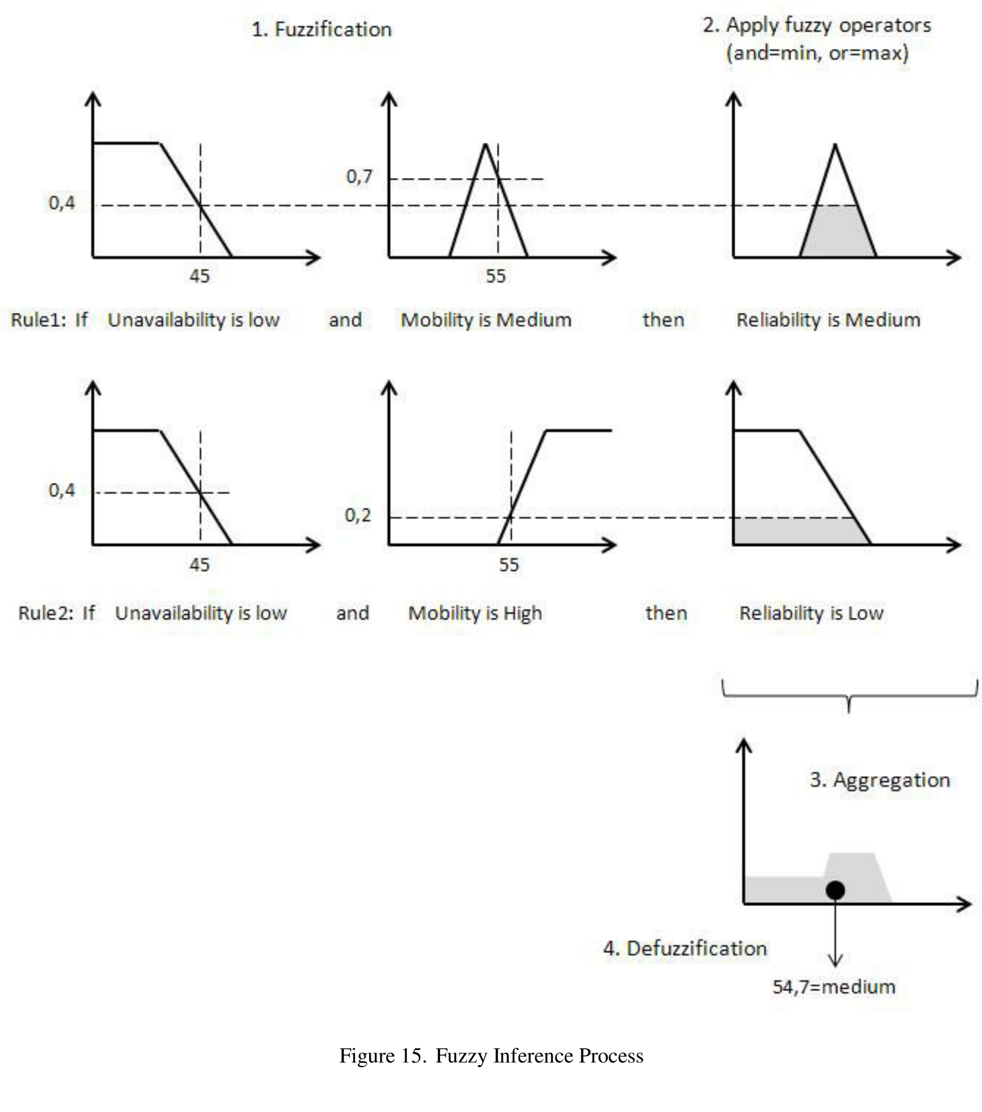


Figure 15. Fuzzy Inference Process

3- Aggregation: in this step, the system aggregates all output values. The output of the aggregation process is one fuzzy set for each output variable.

4- Defuzzification: The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. The results is centroid of area of aggregated output: (for more information see appendix A)

$$\text{Reliability} = \frac{\int_{z} \mu_A(z) z dz}{\int_{z} \mu_A(z) dz}\text{; where } \mu_A(z) \text{ is the aggregated output} \quad (5)$$

In our example the output is 54,7% which is the Reliability of CH.

3.6 How does FLoR work?

In a traditional routing algorithm, all the nodes of the network are equal. That means for example, the system does not have an idea about their unavailability rate; a node that failed 10 times and a node that crashed 5 times in the same period, are equal. But by using our algorithm, we will color the network by giving a *confidence* level to the nodes.

In our algorithm, each node of the network has a neighbor table which has one row for each neighbor. In a dynamic network, the number of rows changes because of the mobility of the nodes. In this table we keep five parameters for each node: *Energy*, *Unavailability*, *QoL, Mobility* and *confidence*. Each node runs a fuzzy function to compute the *confidence* parameter. The confidence parameter is a dynamic parameter which means it can change from *Low* to *High* and also *High* to *Low*, because the failure parameter is dynamic. Each time the conditions of a neighbor change, the node will update *Confidence* of the neighbor.

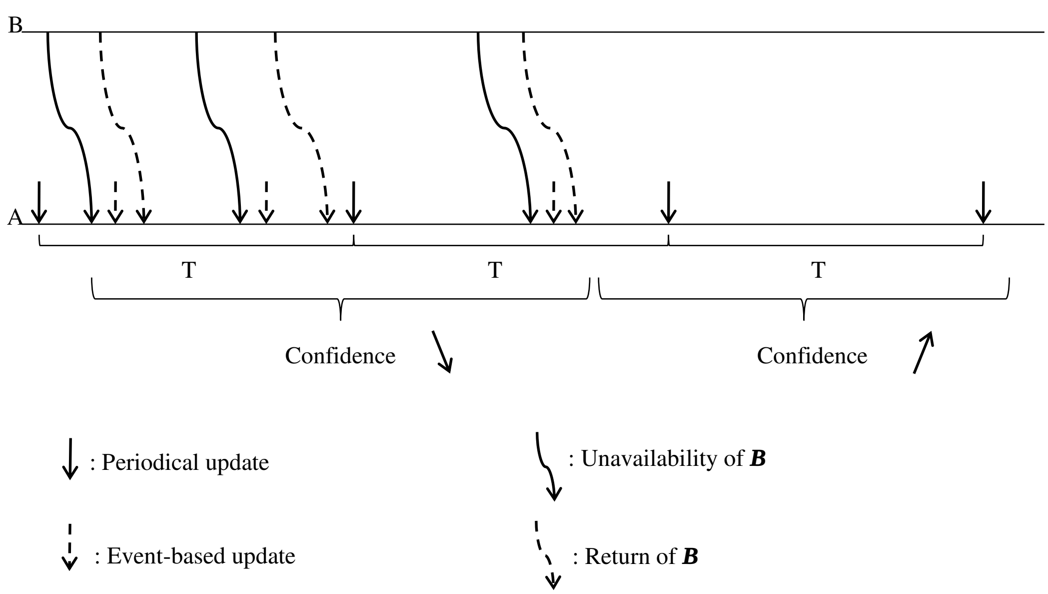


Figure 16. Updating ***Confidence*** parameter

To re-compute the *Confidence* and so updating the confidence, there are 2 methods (Figure 16):

- Periodical update: In each network, due to the mobility or unavailability rate of the nodes and the size of the network, base station will define an update period, in which the *Confidence* of the neighbors will be recomputed.
- Event-based update: Each time that a node detects a change in its neighbor, it re-computes its *Confidence* parameter and updates confidence of the node. For example, in each node A, *Unavailability* of its neighbor B will be computed by: *Unavailability* = fuzzy (n / L); where fuzzy is a function to convert decimal value to fuzzy value, n is number of B's unavailabilities and L is the life time of A. by updating the *Unavailability,* the value of *Confidence* will be recomputed.

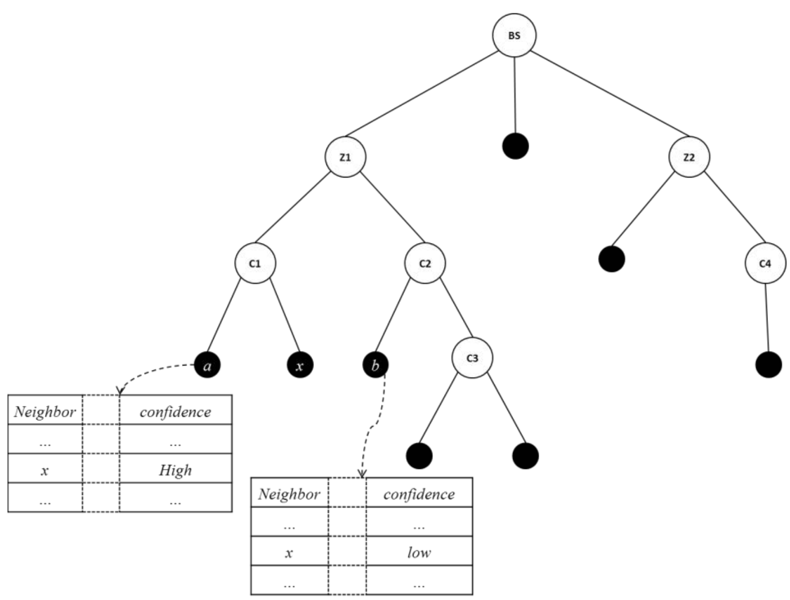


Figure 17. A senario of Neighbor Tables

Figure 17 shows an example of neighbor table in a sample network. We have the nodes '*a*' and '*b*', and their common neighbor, '*x*'. We can see in this figure that *'x'* has different levels of *confidence* in neighbor table of '*a'* and '*b'*. As we said, we use 4 parameters to compute *Confidence* of each node: its *Energy* level, its *Mobility*, its *Unavailability* rate and *QoL* . The *Energy* level and *QoL* of *'x'* will be the same in neighbor table of '*a'* and '*b'*,. But, *Mobility* and *Unavailability* will be different. '*a'* and '*b'* may detect different number of *Mobility* and *Unavailability* for '*x'*, because of difference of environmental conditions between the nodes. Therefore in our proposal, even a common neighbor of two nodes, may have different level of *confidence* in each of them.

3.6.1 Different processes in the nodes

This section presents the process in the nodes to answer different messages on them. In Table II, we find action of a ZH or a CH to answer *Join* message from a node. First, the ZH (or CH) will check if *n* is in the {Unknown} or not, if *n* is in this state, it will remove *n* from the set, also if *n* is not its neighbor, *n* will be added into { myNeighbors }.

Then the ZH will verify *admission condition*: If OK, it will accept *n* as a new member; If KO, and *n* hasn't any other choice, ZH sends a OK message to *n* and chooses one of his actual members with smallest QoL than *n* and will send a request to join another one and will wait for a response, if the answer is KO it will choose another child node to send *Join_Other*.

Table III shows that when a node revives *Invite* from a CH or a ZH, *c*, it will add the node in its {myNeighbors} set and then if it has not a parent, it will send a *Join* message to *c*. If the node has already a parent, it will run the Fuzzy decision making function to know if c is better than its current parent or not: If c is better it will choose it by sending a Join message to it and by receiving OK from c, it sends also a Quit message to its current parent.

TABLE II. RECEIVE JOIN IN A ZH OR A CH

```
Receive (Join from n)
  accepted = False;
  If   n∈ {Unknown} then {Unknown}={Unknown} – n;
  If  n∉{myNeighbors} then
          {myNeighbors}={myNeighbors}+ n;
  If (admission condition is OK) then  accepted = true;
  Else // admission condition is KO
      If (Join.NumberOfCandidates is1) then  accepted = true;
       For  c ∈ {myChild} |
                    (c.QoL is Min) & (n.QoL > c.QoL)
             Send (c,  Join_Other);
             Wait (receive (msg, c));
             If (OK or Timeout)
                 accepted=true;
                 {myChild}={myChild}-c;
                 If (Timeout)
                     {Unknown}={Unknown} – c;
                     {myNeighbors}={myNeighbors}- c;
 If ( accepted == true)
       {myChild}={myChild}+ n;
       Send (n, OK);
 else
     Send (n,  Join_Other);
```

TABLE III. RECEIVE INVITE FROM A ZH OR A CH

```
Receive (Invite from c)
         If (this.CH==null)
              Send (c, Join)
              Wait (receive (OK from c);
         Else // this.CH ≠ null
                Best = FuzzyFunction (this.CH, c);
                If (Best == c)
                      Send (c, Join);
                      Wait (receive (OK from c)
                      If (OK)
                          Send (this.CH, Quit);
                          Join c;
```

TABLE IV. RECEIVE JOIN-OTHER FROM A ZH OR A CH

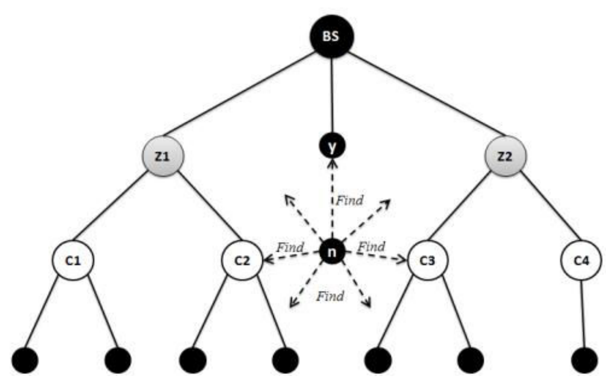
```
Receive (Join-Other from n)
        Found=false;
        For all m ∈ {myNeighbours}
              Send (m, Join);
              Wait (receive (OK from m)
              If (OK)
                   Found=True;
                   Break;
        If (Found = true) then
              Send(n, OK)
              For all c ∈ {myNeighbours} - m
                     Send (c, Hello);
        Else
              Send(n, KO)
```

Table IV shows that in a node, by receiving Join-Other message, the Node will send Join message to its neighbors and if it receives OK, from one of them, it will send OK to its parent node, if no, it will send KO.

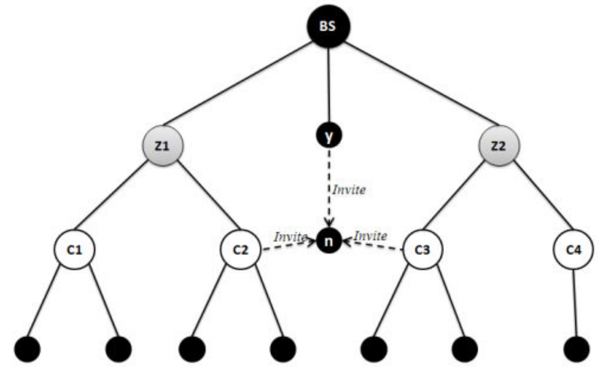
3.6.2 Some examples

In this section we will present our protocol with some examples. Let's start with a simple example. Table V shows the first example. In this network node *n* searches a parent's node. Here are the different steps of the procedure:

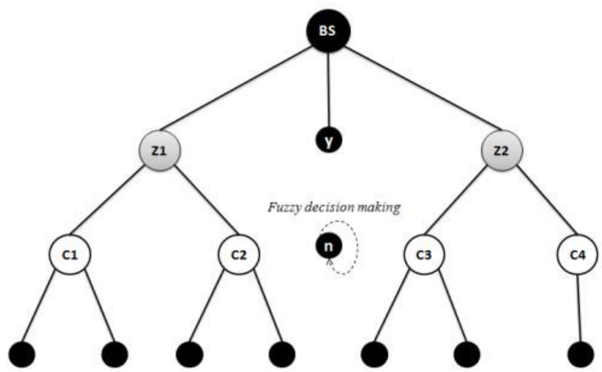
TABLE V. EXAMPLE 1



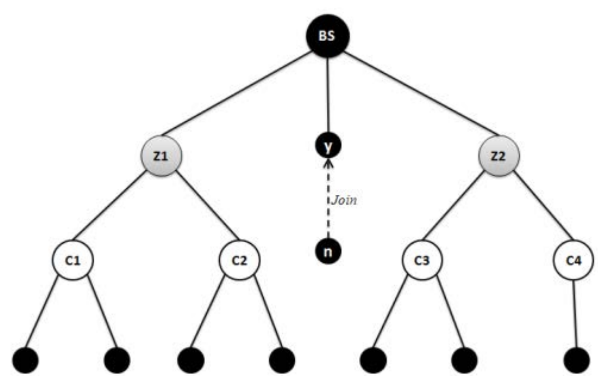
i. n diffuses Find message.



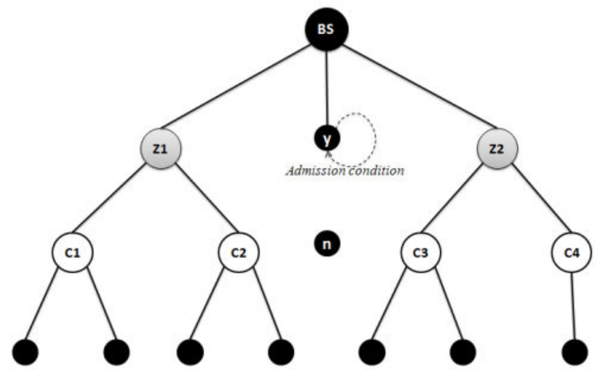
j. Its neighbors, y, C2 and C3 send an Invite message as answer. By the Invite message, they send their Energy, QoL.



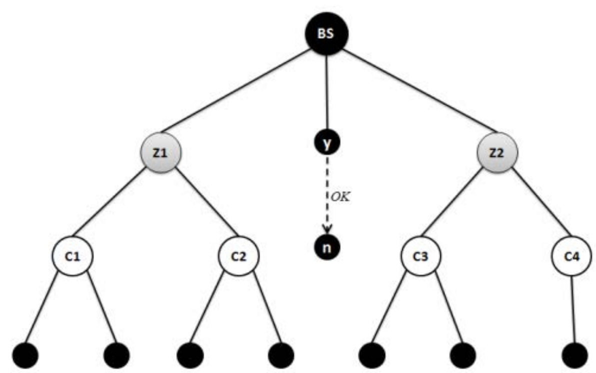
k. n runs a Fuzzy function to evaluate the candidates.



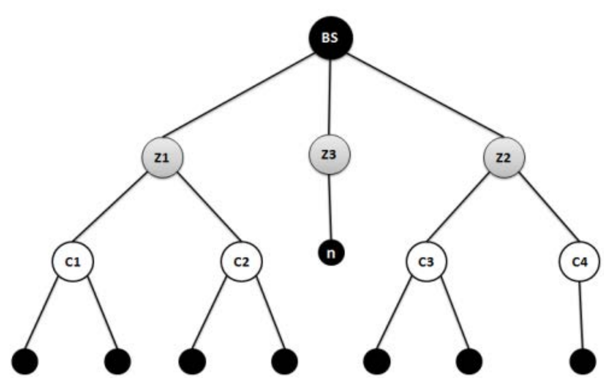
l. y was the best node to be parent of n. therefore, n sends a Join message to y.



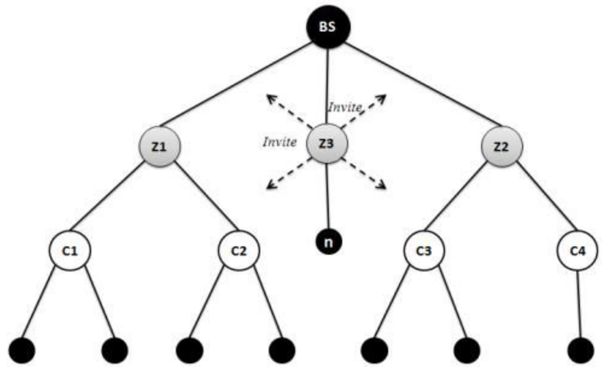
m. By receiving Join message, y verifies Admission condition.



n. Admission condition was ok, therefore y sends a OK message to n.



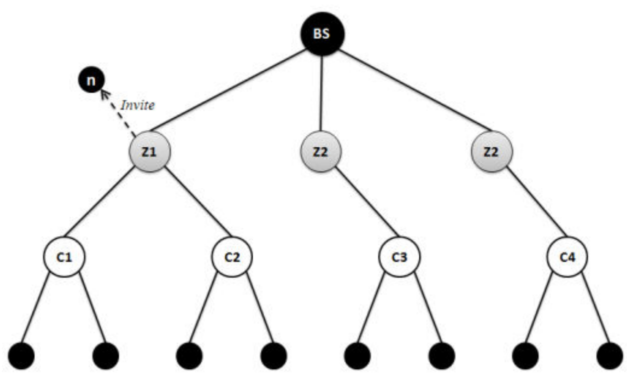
o. n joins y, and the state of y will be changed to a Zone head.



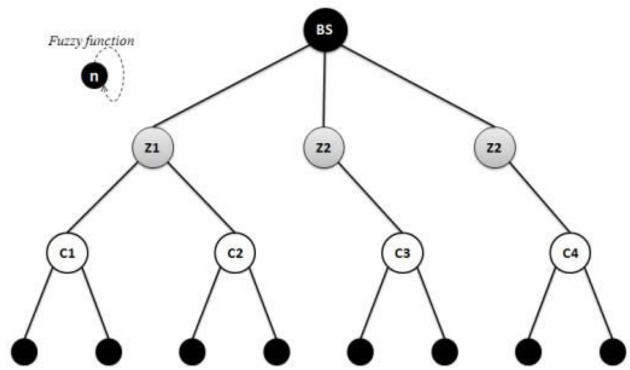
p. As the state of the y is changed, y diffuses a Hello message to announce this.

As second example Table VI shows a scenario with a node that has one possible choice. In this example n diffuses Find message and receives just one invite message from Z1.

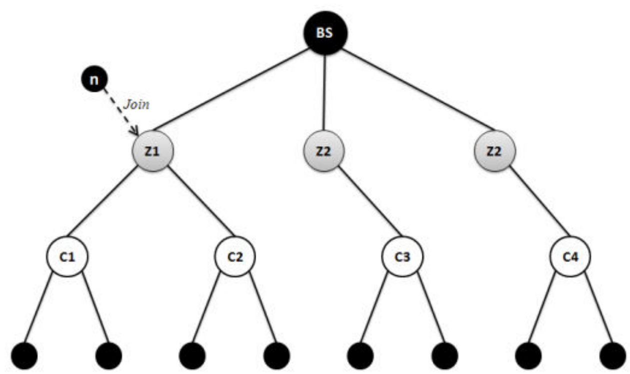
TABLE VI. EXAMPLE 2



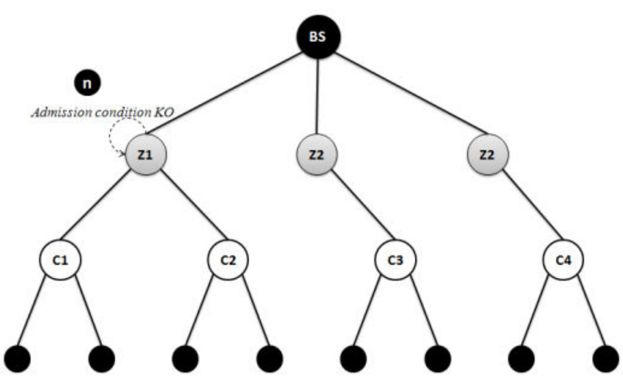
a. Z1 (a zone head) that received the Find message sends an invite message as answer.



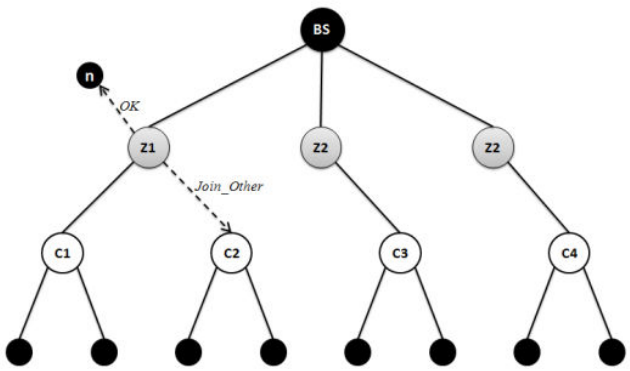
b. n received just one invite message, that means it has no other choice. It runs the fuzzy function to compute QoL. Therefore it sends a join message to Z1.



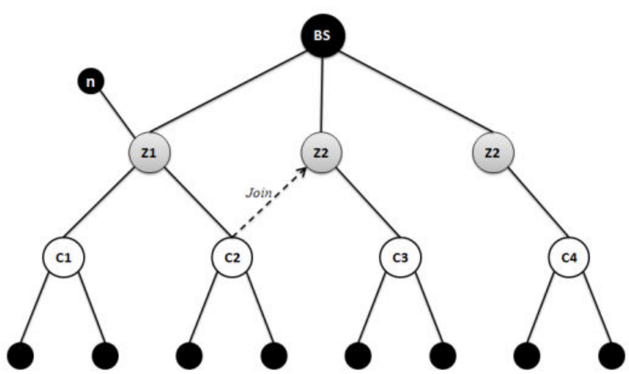
c. n sends a Join message to Z1.



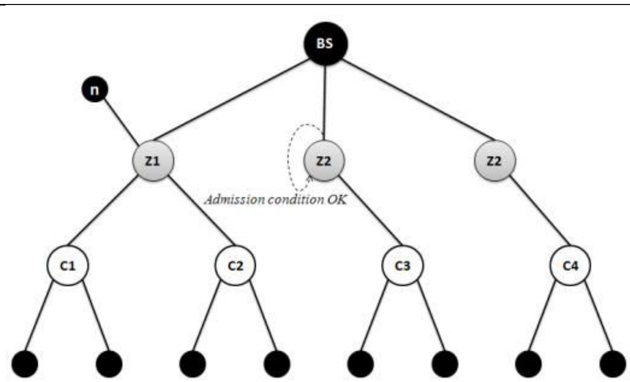
d. By receiving Join message, Z1 verifies Admission condition. But it is not OK.



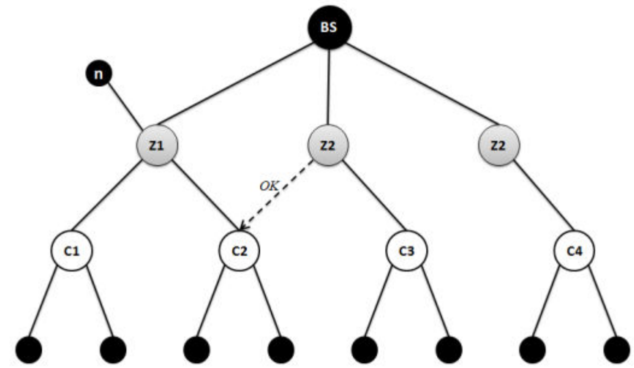
e. As n hasn't any other choice, Z1 sends a OK message to n and a Join_Other message to C2, one of its child.



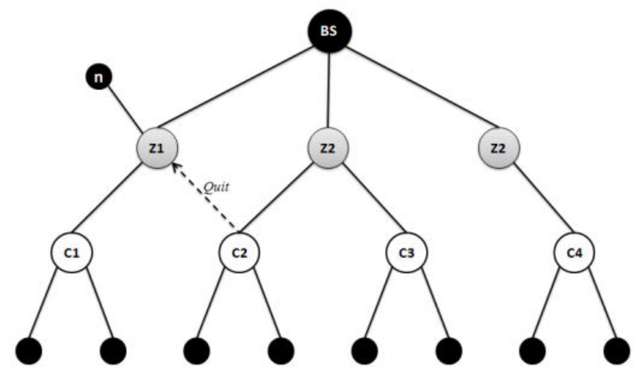
f. n joins Z1, and C2 sends a Join message to Z2.



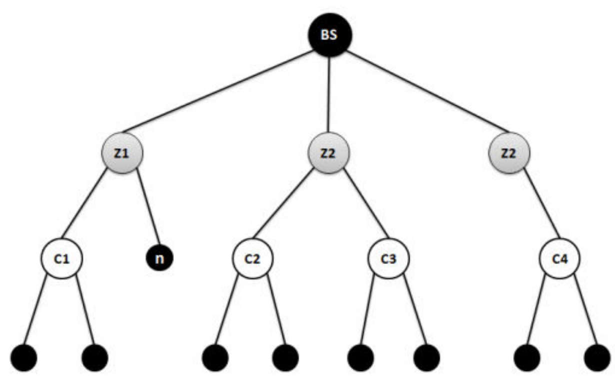
g. Z2 verifies Admission condition which is OK in this example.



h. Z2 sends a OK message to C2.



i. C2 joins Z2 and sends a Quit message to Z1. By receiving Quit message, Z1 delete C2 from its Child set.



j. This figure shows final cluster of the network.

3.7 Evaluation

This section presents the evaluated performance of FLoR. we compared FLoR with AODV [54] and LEACH [35].

3.7.1 Overview of AODV and LEACH

3.7.1.1 AODV

AODV (Ad hoc On-demand Distance Vector) [54] is based on on-demand routing algorithm. In this protocol, routes are built between nodes only as desired by source nodes. And it uses traditional routing table, one entry per destination, to maintain routing information.

AODV uses sequence numbers to ensure the freshness of routes, and also uses a timer-based route expiry mechanism to promptly discard stale routes. It sends Hello message periodically to detect and monitor links to neighbors, without the support of MAC layer. Moreover, destination node and any node that has a current route to the destination node can generate the reply message.

3.7.1.2 LEACH

LEACH (Low-Energy Adaptive Clustering Hierarchy) is a family of protocols developed in [35]. LEACH is a good approximation of a proactive network protocol, with some minor differences. Once the clusters are formed, the cluster heads broadcast a TDMA schedule giving the order in which the cluster members can transmit their data. The total time required to complete this schedule is called the frame time TF.

Every node in the cluster has its own slot in the frame, during which it transmits data to the cluster head. When the last node in the schedule has transmitted its data, the schedule repeats. The report time discussed earlier is equivalent to the frame time in LEACH. The frame time is not broadcast by the cluster head, though it is derived from the TDMA schedule. However, it is not under user control. Also, the attributes are predetermined and are not changed midway.

3.7.2 Evaluation metrics - QoS requirements in wireless sensor networks

Several protocols and algorithms have been proposed for QoS routing in sensor networks. However, the nature of sensor networks poses unique challenges compared to general wireless networks and, thus, requires special attention. The type of target application can play an important role for QoS in wireless sensor networks. QoS in wireless sensor networks can be characterized by reliability, timeliness, robustness, availability, and security among all others.

The throughput, delay, Network lifetime, packet delivery rate and Routing overhead are the most fundamental parameters and may be used to measure the degree of satisfaction of these services. This section explains their descriptions.

3.7.2.1 Delivery ratio

The packet delivery ratio is measured as the ratio of the number of data packets delivered to the base station to the number of data packets sent by the source sensors.

3.7.2.2 Throughput (Kbit/sec)

Throughput is the effective number of data flow transported within a certain period of time, also specified as bandwidth in some situations. In general, a bigger throughput shows good performance of the network. Those nodes that generate high-speed data streams, such as a camera sensor node used to transmit images for target tracking, often require high throughput. In order to improve the resource efficiency, furthermore, the throughput of WSAN should often be maximized.

3.7.2.3 Routing overhead

The routing overhead measures by the total number of control packets sent divided by the number of data packets delivered successfully. The overhead of cluster based routing can be associated with one of the following operations: Route Discovery, Route Maintenance and Cluster Maintenance.

3.7.2.4 Delay (ms)

Delay is the time elapsed from the departure of a data packet from the source node to the arrival at the destination node. This includes the processing delay, queuing delay, transmission delay, and propagation delay. The processing delay is the time sensors take to process the packet header, while the queuing delay is the time the packet stays in sensor queues.

The transmission delay is the time it takes to push the packet's bits onto the link. The propagation delay is the time it takes for the signal to propagate through the transmission medium. Delay sensitive applications usually require the delivery of data packets in real time. Notice that real time does not necessarily mean fast computation or communication.

3.7.2.5 Energy consumption

Energy consumption is one of critical challenges in wireless sensor networks since sensor nodes are typically powered by batteries with a limited capacity. In order to increase the lifetime of sensor networks, it is important to use the protocols and algorithms to ensure that each node conserve energy as much as possible.

3.7.3 Simulation environment

We use OPNET simulator to evaluate FLoR. Table VII, shows our simulation environment's parameters.

The evaluated parameters are: ZHs' load, Average QoL in the zones, Packet delivery ratio, Throughput, Routing overhead, Delay and Energy consumption.

To evaluate ZHs' load and the average QoL in the zones, the network consists of 100 nodes and for the other parameters the number of the nodes was varied from 25 to 100.

The simulation consists of homogeneous scattered randomly within a 500x500 m surface which presents typically surface of a "Maison Vill'Âge" housing scheme. The Base Station (BS) is located at the center of the surface (250,250).

To reduce simulation complexity, to have clear results, we consider that the network consists of several low mobility wireless nodes, just 20% of the nodes are mobile and their speed is 0.5 m/s with random direction.

For simplicity, we refer to each time unit as a round. Transmission and sensing range are 15 m and 8 m respectively.

Packet size is 128 bytes and energy consumption is 10nJ/bit [55]. Data transmission rate is 15 Packet/s. We use random mobility and failure model. Each node has 0.5J initial energy [35] and the energy consumption due to communication is calculated using the first order energy model [35].

When a node sends a message to its neighbor or its parent, it waits an acknowledge message. If it doesn't receive this message after an authorized number of retries (specified by MAC protocol) the node will be considered unavailable. The number of the faulty nodes is random and is in maximum 20%

of the nodes. When a node fails or crashes, it is not dead; it will return to the network after a variable time, $t \neq \infty$. During the simulation, this time is chosen randomly.

TABLE VII. SIMULATION PARAMETERS

Node Number	25, 50, 75, 100
Surface	500m x 500m
Transmission range	15m
Data transmission rate	15 packet/sec
Failure model	Random
Packet size	128 bytes
Initial Energy	5J
Energy consumption (Calculation, receive and send)	10 nJ/bit

3.7.4 Simulation results

This section presents the evaluation results of FLoR. The first part presents the clustering performance of FLOR and second part presents the routing performance by comparing FLoR with AODV and LEACH.

3.7.4.1 Clustering performance

In this section we evaluate ZHs' loads and the mean of the Zone loads as parameters to evaluate the clustering performance of FLoR.

3.7.4.1.1 ZHs' load

Figure 18 shows load in ZHs of the simulated network. We find in this figure that after 10 rounds, network has 6 zones and load of ZHs are between medium and high. After 100 rounds network has 7 zones and ZHs have a load between medium and low. After 200 and 300 round, the network has 8 zones and the average of load in ZHs after low. These results show that there is a convergence to an optimal number of zones and the optimal value of the load (Low). Optimal value of the ZHs seems to be 8 in our simulation scenario. Note that in our algorithm, a ZH is a node with one hop distance with BS; therefor the number of ZHs stays a limited number.

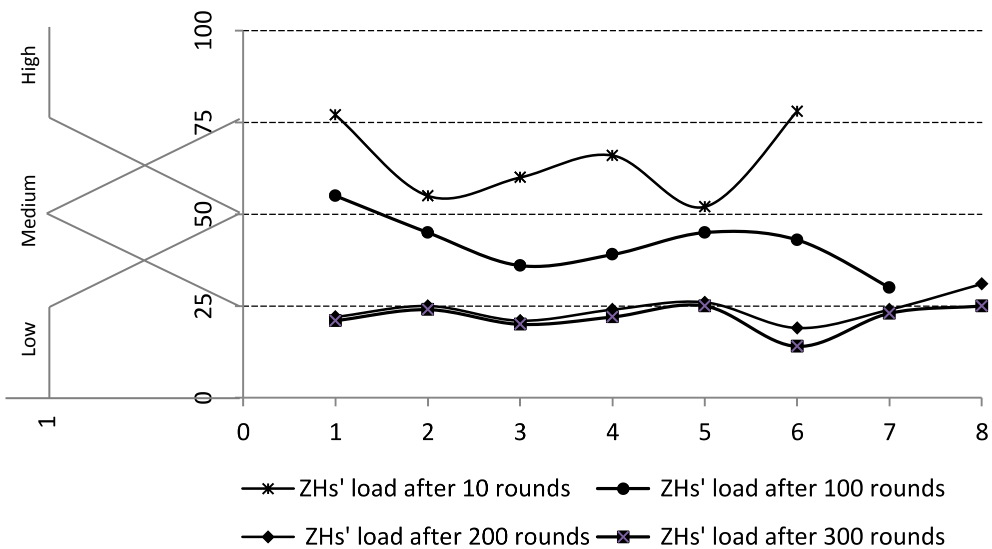


Figure 18. Network's ZHs' load

3.7.4.1.2 Average QoL in the zones

As we explained before, QoL in a zone shows the connectivity of the nodes. A QoL with value of high shows a good connectivity between the nodes and a low QoL is results of unstable connection between the nodes.

Figure 19 shows average QoL in the zones of the simulated network. We find in this figure that after 10 rounds, network has 6 zones and average QoL in the zones in between medium and high, and after 100 rounds network has 7 zones with average QoL near to high.

The figure shows that the average value of QoL in the network goes toward its optimal value (High). These results show the efficiency of our protocol to establish reliable and stable connections between the nodes.

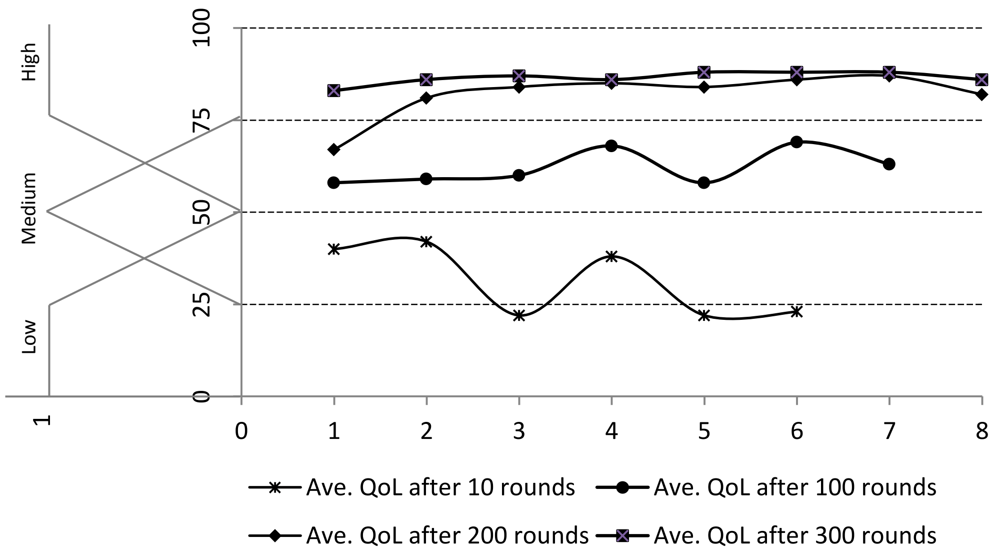


Figure 19. Average QoL in zones

3.7.4.2 Routing performance

To evaluate the routing performance of the FLoR, we use Packet delivery ratio, Throughput, Routing overhead and the Delay.

3.7.4.3 Packet delivery ratio

In this section, we focus on the packet delivery ratio. We can find the simulation results in figure 20.

The simulation shows that our protocol increases the data delivery in the network and therefore it greatly adapts mobility and unavailability of the nodes.

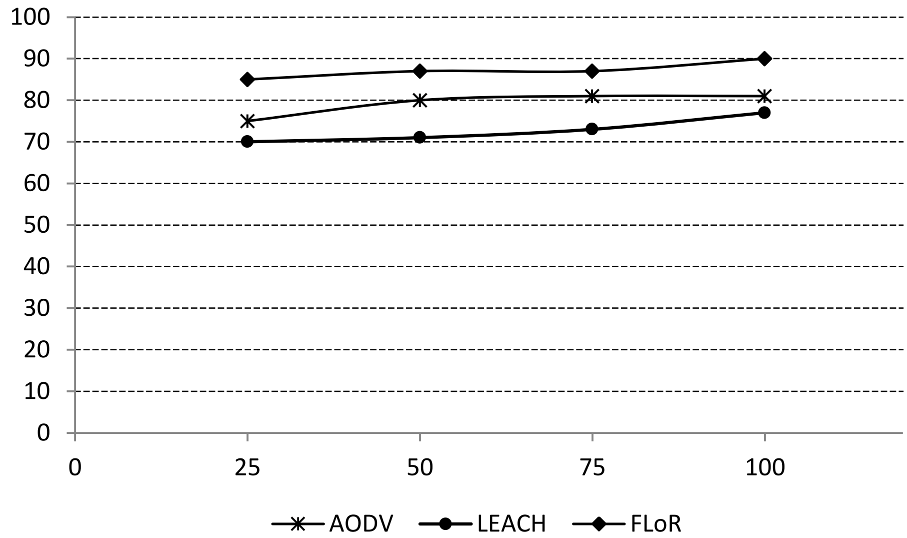


Figure 20. Delivery ratio

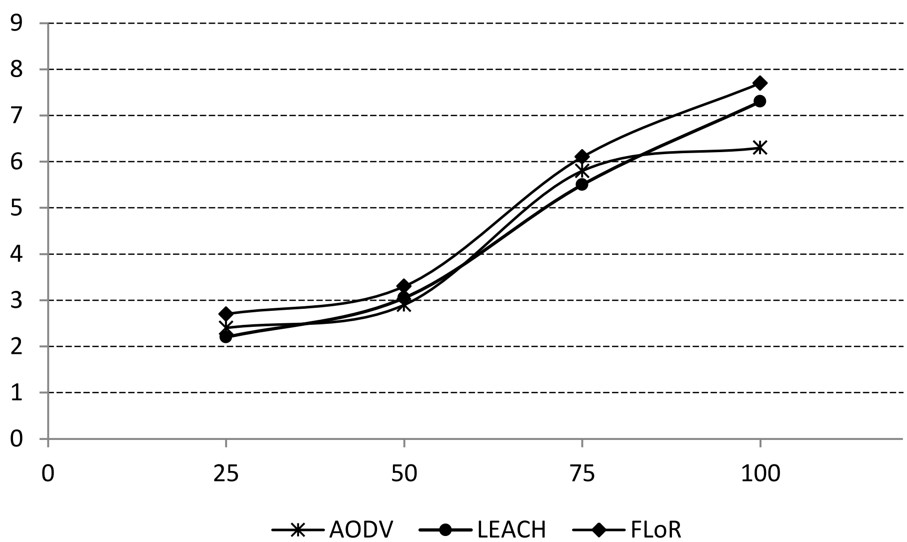


Figure 21. Throughput

3.7.4.4 Throughput

Based on figure 21, we observe in LEACH, the throughput is lower than the other but increases with increase in network size. FLoR shows a better performance than the others and in FLoR also the throughput increases with increases in node numbers.

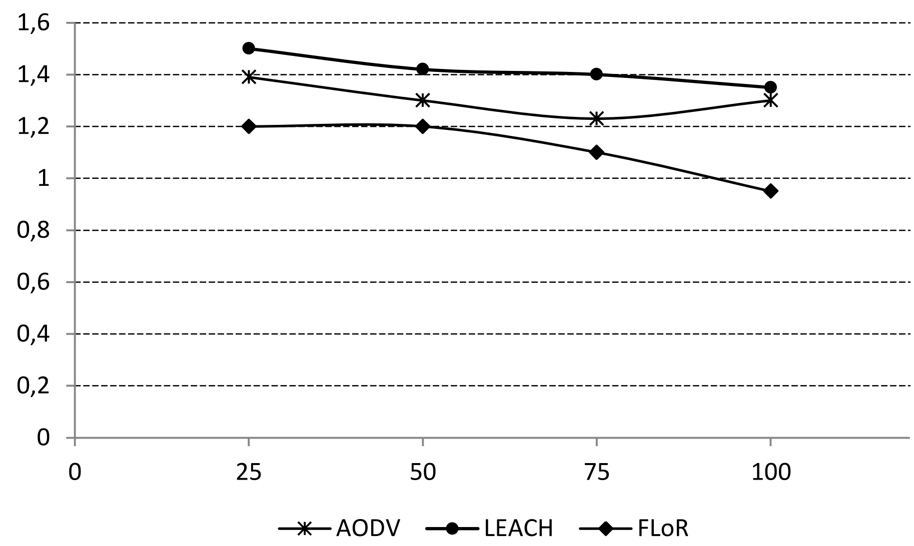


Figure 22. Routing overhead

3.7.4.5 Routing overhead

From figure 22, in AODV, the result of routing overhead increases at some points and decreases at other points like packet delivery rate. In LEACH, the routing overhead decreases smoothly when the number of nodes increases. The performance of our proposal is better with more number of nodes than in comparison with other two protocols.

3.7.4.6 Delay

Figure 23 shows the comparison of delay. FLoR, the average end-to-end delay become very low and increases when the number of node increases, but slowly and better as compare to others.

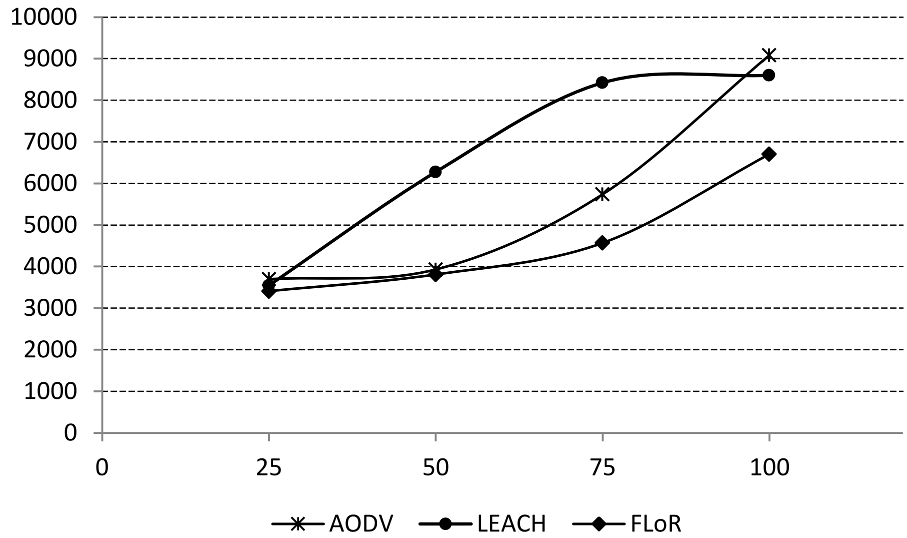


Figure 23. Delay (ms)

3.7.4.7 Energy consumption

Figure 24 compares the energy consumption of FLoR, AODV and LEACH. As this figure shows, FLoR reduces energy consumption of the nodes when the network has 75 and more nodes. When the network is composed of 25 sensors, LEACH has the lower energy consumption and FLoR has the worse energy consumption. In a network with 50 nodes, LEACH has the best energy consumption and AODV is the worse one.

The figure shows that by increasing the number of nodes of the network, energy consumption of FLoR decreases, in contrast in LEACH and AODV energy consumption decreases by increasing the number of the nodes of the network.

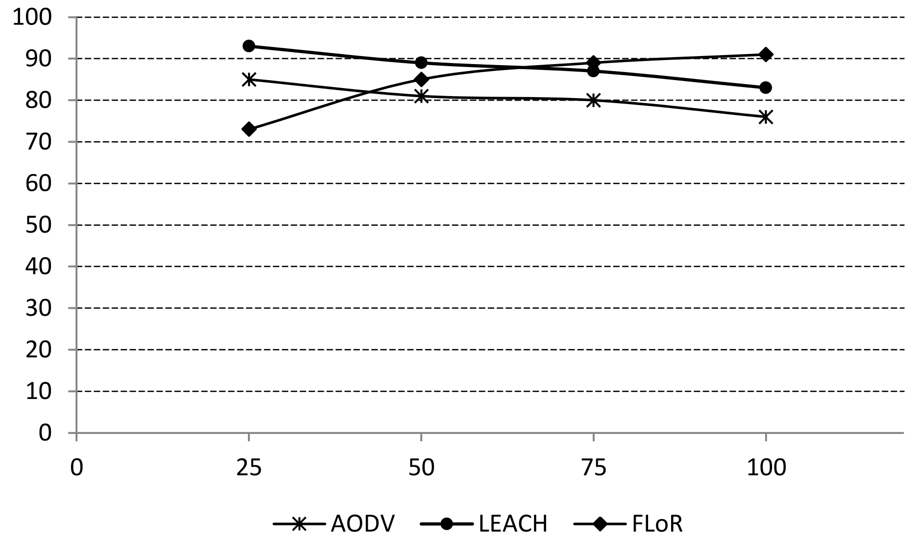


Figure 24. Energy consumption

3.8 Conclusion

In this chapter FLoR, an adaptive clustering and routing algorithm was proposed. FLoR uses fuzzy logic to form the cluster dynamically (See figure 25). Clusters are formed dynamically to adapt to the network changes. Its admission policy is based on the routing feasibility of the system under this algorithm. All decision making process are based on Fuzzy logic.

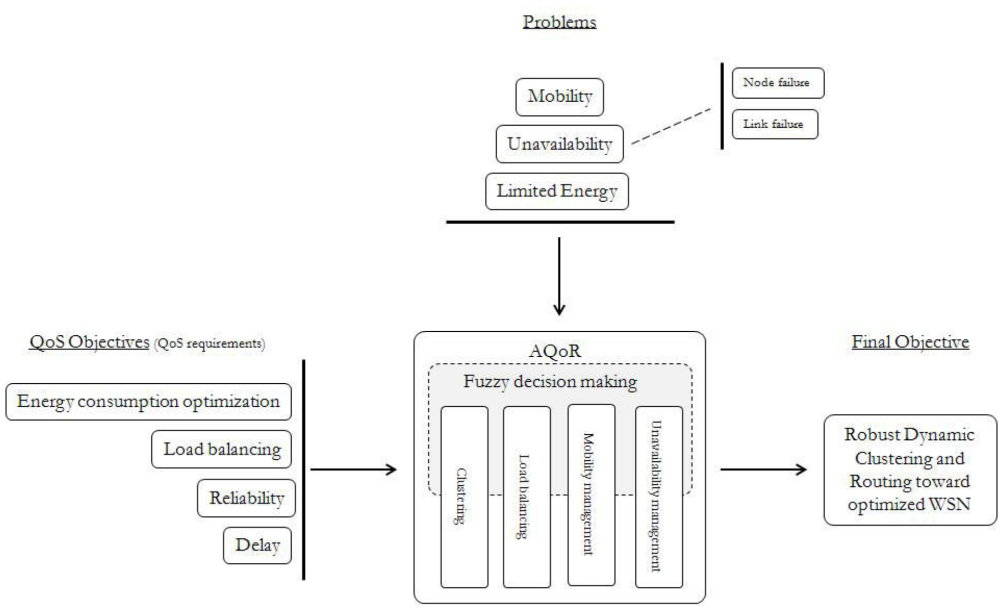


Figure 25. FLoR

In order to provide a robust dynamic clustering and routing by considering QoS objectives, FLoR uses *Energy*, *Mobility*, *Unavailability history* and *Load* of the nodes which are some new definitions that were introduced in this algorithm. Mobility has a new definition in this algorithm; Mobility of a node is its rate of parent changes. To manage the node and link failure, FLoR uses Unavailability, which is the rate of unavailability of a node. Stable route recovery, high data delivery ratio and good load balancing are the main characteristics that the proposed algorithm adds to the ad-hoc sensor networks.

Chapter 4 CodaQ

a Context-aware Adaptive QoS Middleware

For providing real-time data collecting in a telehealthcare system composed of wireless sensor network and home automation network, a middleware called CodaQ is designed. It provides context data and takes into account QoS requirements of the applications. This chapter presents its architecture and we show how the data are modeled for including context information and how the QoS requirements are handled within a middleware. Measurements on a test bed have been carried out showing the good performance of our design.

4.1 Introduction

Activity monitoring (or actimetry) of the older people at home is one of the important applications of the telehealthcare systems. Activity monitoring for healthcare can range from very simple alerts like the high rate of using WC, to a complex intelligent system to analyze the activity patterns of the person in a long period. The aim is to determine any abnormal behavior in the daily life and any hidden health problem and risks like diabetes. Actimetry needs to collect in real-time sensor data through an efficient communication system with QoS guarantee. Wireless sensor network (WSN) is an emerging technology for building telehealthcare systems and has inspired important research efforts [56]. One of the main research topics is to provide the QoS for the sensor data transfer in terms of timeliness and reliability under stringent energy constraints. However how to map the application requirement and the network QoS services needs still further investigation.

In a typical activity monitoring system data traffic flows can be classified into 3 categories with different priorities: 1) Medical; 2) Environmental and Home automation sensor and actuator; 3) Multimedia. Medical data flow has the highest priority and multimedia the lowest one. Data collection can be achieved through a home gateway. Different data collection strategies can be applied in each category: Event-based, periodic or mixed data collection, which demand managing the QoS requirements (by applying priorities) and temporal coherence (time stamp). The QoS requirements of the sensor data are determined by sensor data type and the condition of the patient. For example, the medical information like ECG may require a higher priority than the other sensors when a patient has an unusual heart rate, in this case the home gateway must be able to give the higher priority to ECG data flow. The temporal coherence is needed for multi-sensor data fusion (e.g. related data are collected during a coherent time window) and this can be achieved by time stamping at the home gateway. Moreover sensor data should be interpreted within their current context. This can be achieved by using context-aware computing [57, 58]. Context-aware computing is shifting the healthcare from treatment by professionals in hospitals to self-care, mobile care, home care and preventive care. Context-awareness refers to knowledge and understanding of the surrounding environment within which the decision support system has to operate.

In this PhD work we developed a context-aware middleware for our telehomecare and activity monitoring system. This chapter presents its architecture.

4.2 Definitions

4.2.1 Context

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves [57].

This definition makes it easier for an application developer to enumerate the context for a given application scenario. If a piece of information can be used to characterize the situation of a participant in an interaction, then that information is context.

For example, the user's location is a context, because it can be used to characterize the user's situation. If the user is located in the United States, the sum of the weights will be presented in terms of pounds and ounces. If the user is located in France, the sum of the weights will be presented in terms of kilograms. Therefore, the user's location is context because it can be used to characterize the user's situation. [58]

4.2.2 Context awareness and self-adapting

A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task to work toward to the automation of a software system based on knowledge of the user's context.

With progress of pervasive computing technology, context-aware applications are more and more widely used. Recent research on context-awareness mainly focuses on two aspects: the methodology of modeling and the mechanism of awareness process. Context-aware computing has many features which help us to develop telehomecare systems.

The first feature is contextual sensing, which is the ability to detect contextual information and present it to the user, augmenting the user's sensory system. This is similar to proximate selection, except in this case, the user does not necessarily need to select one of the context items for more information (i.e. the context may be the information required) [57]. The next feature is contextual adaptation and is the ability to execute or modify a service automatically based on the current context. The third feature, contextual resource discovery, allows context aware applications to locate and exploit resources and services that are relevant to the user's context. This third feature maps directly to automatic contextual reconfiguration [57]. The final feature, contextual augmentation, is the ability to associate digital data with the user's context. A user can view the data when he is in that associated context. For example, a user can create a virtual note providing details about a broken television and attach the note to the television. When another user is close to the television or attempts to use it, he will see the virtual note left previously [57].

The concepts of context awareness and self-adaptation are often sources of confusion, because self-adaptive applications often adapt their behavior based on the context and therefore, it is often difficult to make a clear distinction between these two concepts [58]. A purely context-aware application is identified as using the context information simply to complement its functional behavior. On the other hand, a context-adaptive application adjusts its extra-functional behavior based on the context information. All self-adaptive applications can be viewed as context-adaptive, because they use context information for adapting their extra-functional behavior [57].

4.2.3 QoC – Quality of Context

Quality of Context (QoC) is any information that describes the quality of information that is used as context information. Thus, QoC refers to information and neither to the process nor the hardware component that possibly provide the information [59].

4.2.3.1 Indicators of QoC

There are some indicators to evaluate the QoC in a context-aware middleware. Here we will describe some of them based on [60].

4.2.3.1.1 Precision

It is defined as the granularity with which context information describes a real world situation. Context precision specifies a range of values associated with a context attribute value. For example, for location information retrieved from a GPS sensor, the precision is usually a constant value, e.g. approximately five meters.

4.2.3.1.2 Freshness

In context-aware systems, all sensors stream time varying measurements. In these systems the freshness of the messages must be guaranteed. Context data freshness implies that the data is recent, and it ensures that no adversary replayed old messages. In a context-aware system, the freshness is the time that elapses between the determination of context information and its delivery to a requester. This is an important requirement as context that is too old may not be useful to a requester. Also, the freshness of context is directly proportional to its privacy sensitiveness, i.e. users may restrain recent context information from being provided to applications. The system would evaluate application requirements and user privacy preferences regarding freshness to decide on its caching strategy. Quantifying freshness is a relatively simple task as it represents a period of time.

4.2.4 Middleware

Middleware is defined as software systems which provide an abstraction layer between an operating system and applications running in a distributed environment [61]. The role of middleware is to provide an additional layer of abstraction suitable for a specific type of applications. Middleware is normally intended for a specific type of distributed system, and even though a given middleware systems claims to be suitable for general distributed systems, underlying assumptions often implies certain limitations for the usefulness of the system. In traditional distributed systems, the goal of the middleware has been to hide heterogeneity and distribution by providing ways of treating remote resources as if they were local. In wired, static environment, this has proven useful, but in dynamic, wireless environments it breaks down, since applications often need to base decisions on information about distribution and the environment. Instead, middleware systems for context-aware computing focus on providing suitable abstractions for dealing with heterogeneity and distribution without hiding them, and in some cases even provide information about distribution and heterogeneity as context information.

4.2.5 Context ontology

Ontology is a vocabulary representation of context. It is necessary to perform an ontological analysis of the context to clarify its structure, and then applications can work together and interpret contexts based on their knowledge. It is necessary therefor to have standard context ontology [62].

4.3 Challenges in designing context-aware middleware

Context-aware systems represent extremely complex and heterogeneous distributed systems, composed of sensors, actuators, application components, and a variety of context processing components that manage the flow of context information between the sensors/actuators and applications [56].

The need for middleware to seamlessly bind these components together is well recognized. Numerous attempts to build middleware or infrastructure for context-aware systems have been made, but these have provided only partial solutions; for instance, most have not adequately addressed issues such as mobility, fault tolerance or privacy [63].

A change in context provision must be detected quickly and adaptation occurs swiftly. This is particularly important in health-related applications that monitor a person with health problems, as these usually require prompt responses to changes in the health condition of the person [64].

The technological landscape may be summarized as follows. Middleware systems assume that component nodes have significant processing and communications capabilities – assumptions that are typically not respected in pervasive computing systems, and perhaps especially in systems intended for use in existing buildings such as are typically encountered in home automation.

Sensor systems often have only simple control and data interfaces, and do not provide an attractive programming platform for complex applications. It is hard to build decentralized applications and hard to handle the inevitable addition of (or failure of) devices in a way that does not require significant application customization [65].

Generally, we can name these challenges as the important challenges in design of a middleware for sensor networks [57-61]:

- Limited power and resources: Advance microelectronics technology allows tiny devices but limited in energy and resources, i.e. CPU and memory. Middleware should provide mechanisms for efficient processor and memory use while enabling lower-power communication and prolong sensor node lifetime (sleep mode, minimize number of transmission).
- Scalability, mobility, and dynamic network topology: As the application grows, device failure, moving obstacles, mobility, and interference, the network will change frequently. Middleware should maintain performance and robust operation while network changes dynamically. Also, It should support mechanisms for fault tolerance and sensor node self-configuration and self-maintenance.
- Heterogeneity in CPU-power, networking, memory and storage, operating systems: Middleware should be able to interface various kinds of hardware, software and networks.
- Dynamic network organization: In most situations, Client/Server mode is impossible, because no infrastructure is available. Therefore the middleware should support Ad-hoc capability to discover resource and its location which affect the trade-offs among latency, reliability, and energy.
- Real-world integration: Most of the applications are dealing with real-time phenomena and the middleware should provide real-time services.

- Data aggregation: Network generates lots of redundant data, communications cost is much higher than computational cost. Sending a single bit can consume the same energy as executing 1000 instructions. Middleware should able to aggregate data to eliminate redundancy and minimize the number of transmissions to the sink.
- Security: Middleware efforts should concentrate on developing and integrating security in the initial phases of software design, hence achieving different security requirements such as authentication, integrity, freshness, and availability.

To cover all these challenges, an important requirement is a safe and reliable method to acquire the context information. There are three different approaches on how to acquire contextual information [62]:

- **Direct sensor access:** This approach is often used in devices with sensors locally built in. The client software gathers the desired information directly from these sensors, i.e., there is no additional layer for gaining and processing sensor data. Drivers for the sensors are hardwired into the application, so this tightly coupled method is usable only in rare cases. Therefore, it is not suited for distributed systems due to its direct access nature which lacks a component capable of managing multiple concurrent sensor accesses.
- **Middleware infrastructure:** Modern software design uses methods of encapsulation to separate e.g., business logic and graphical user interfaces. The middleware based approach introduces a layered architecture to context-aware systems with the intention of hiding low-level sensing details. Compared to direct sensor access this technique eases extensibility since the client code has not to be

modified anymore and it simplifies the reusability of hardware dependent sensing code due to the strict encapsulation.

- **Context server:** The next logical step is to permit multiple clients access to remote data sources. This distributed approach extends the middleware based architecture by introducing an access managing remote component. Gathering sensor data is moved to this so-called context server to facilitate concurrent multiple access. Besides the reuse of sensors, the usage of a context server has the advantage of relieving clients of resource intensive operations. As probably the majority of end devices used in context-aware systems are mobile gadgets with limitations in computation power, disk space etc., this is an important aspect. In return one has to consider about appropriate protocols, network performance, quality of service parameters etc., when designing a context-aware system based on client-server architecture.

4.4 Reference model

In order to consider the challenges we discussed before, we defined a reference model (Figure 26). This model presents all modules that a Middleware must have to be able to deal with the challenges and ensure a real context deduction.

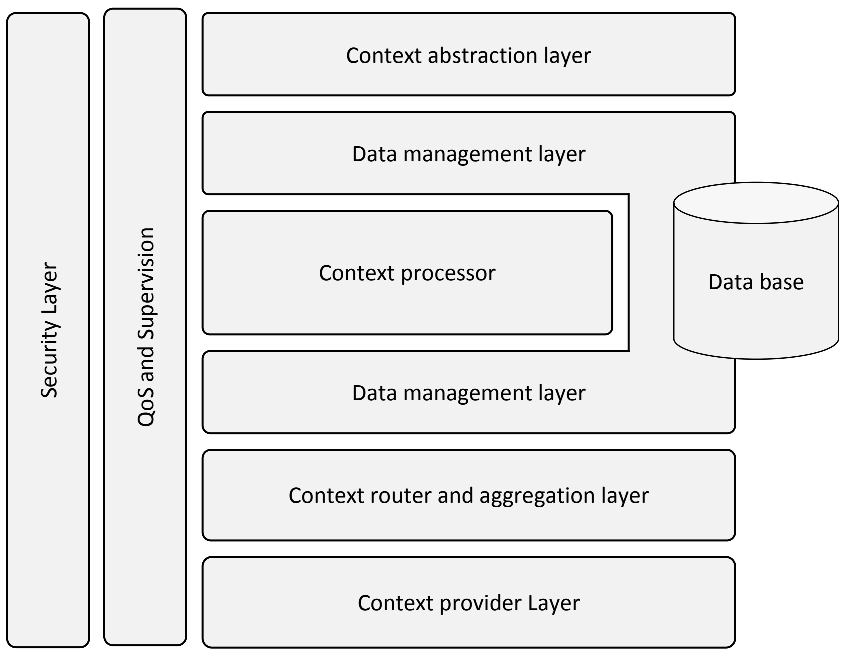


Figure 26. Reference model

In this model, these modules can be found:

- **Context abstraction layer:** provides an abstraction of the context for context-aware application.
- **QoS and Supervision layer:** it guaranties the supervision of the system and applies runtime QoS need.
- **Database layer:** save the events and states detected from the context.

- **Context-processor layer:** this layer use the raw data provided by the context router, to reason the context.
- **Data management layer:** this layer controls the databases and ensures the management of all data that change between the other layers.
- **Context router and aggregation layer:** context router is a gateway between the context provider layer (sensors) and the middleware.
- **Context provider layer:** is a distributed application installed on the sensors to detect and send the events to the middleware.
- **Security layer:** it is the responsible of providing the security of the system.

We will use this model to evaluate the state of the art of context-aware middlewares. The model is used also as a base for our Middleware's (CodaQ) architecture.

4.5 Related works

In [66], the authors present a middleware based on SOA (Service-Oriented Architecture) for context-aware home networks. This middleware addresses design issues such as heterogeneity, dynamicity, and extensibility issues of ubiquitous computing environment. The approach proposes a context model supporting semantic level interoperability of context. The authors target the heterogeneous platforms in order to ensure an interoperable interaction between these platforms. The system uses OWL which is a web ontology language for modeling the context.

Another middleware is presented in [67]. It proposes a Service-Oriented Context-Aware Middleware (SOCAM) architecture for the building and prototyping of context-aware services. The goal is to provide an efficient support for acquiring, discovering, interpreting and accessing various contexts to build context-aware services. The system proposes a formal context model based on ontology using OWL to address issues including semantic representation, context reasoning, context classification and dependency. SOCAM is based on the idea of using ontologies to model context. The model is then used by an interpreter to reason about context.

The SOCAM architecture consists of: *Context Providers*, *Context Interpreters*, *a Context Database*, *a Service Location Service*, and *Context-aware Mobile Services*. *Context Providers* provide external or internal context, which can be used by the mobile services directly or by *Context Provider* to provide higher-order context. Externally, the Context Interpreter acts as a Context Provider. Context is represented as instances of the ontology model. The Context Interpreter consists of a Context Reasoner, and a *Context Database*, which contains instances of the current ontology, either from users or Context Providers.

In SOCAM, the context is updated by a triggering mechanism with different intervals. Context Providers register with the *Service Location Service*, thus allowing Mobile Services to locate them. The *Mobile Services* can obtain context either by querying the located Context Providers, or by registering for specific events. SOCAM supports *rules* for specifying which methods should be invoked on events. The rules are predefined and loaded into the Context Reasoner.

An adaptive middleware design for context-aware applications is presented in [63]. The middleware abstracts the applications from the sensors that provide context. It also aims to implement autonomic properties, such as self-configuration and failure tolerant.

The goal in the design of this middleware has been to provide adaptation with good performance. The approach assumes that Context Providers (CPs) are able to estimate their QoC (Quality of Context: precision of the information provided by middleware about the context). It remains to be seen in practice how well this assumption holds.

[68] discusses the concept of an adaptive middleware with the aim to provide a framework to optimize the shared resources in adaptive homes. To do this, the authors propose the EMISS (Energy Monitoring via the Internet and Sensors for Sustainable living). The principal architecture of EMISS consists of an array of sensors, a transaction server (TS) and a web server (WS). In this architecture, TS manages the sensors and acts as gateway between sensor-enabled devices, it receives also data of active sensors and integrates updates in measurements database.

A context-aware middleware is presented in [65], which has been developed as a part of PACE project (Pervasive, Autonomic, Context-aware Environments.

The middleware has been developed according to the following design principles:

- The model(s) of context information used in a context-aware system should be explicitly represented within the system.
- The context-aware behavior of context-aware applications should be determined, at least in part, by external specifications that can be customized by users and evolved along with the context model (again, without forcing re-implementation of any components).
- The communication between application components, and between the components and middleware services, should not be tightly bound to the application logic, so that a significant re-implementation effort is required when the underlying transport protocols or service interfaces change.

ConStruct is described in [64]. It is a knowledge-based pervasive computing middleware that provides semantically unification over a range of home- and web-based automation systems. ConStruct is a sensor-fusion-based middleware for smart homes. In ConStruct all data are modeled using the Resource Description Framework (RDF) [69], which provides a standardized way in which to model contextual information and properties. ConStruct stores and manipulates this data using the Jena Framework [70].

Aura [71] is a task oriented middleware for infrastructural environments. It provides services for management of tasks, applications, and context. Tasks are controlled by a Task Manager, which handles migration of tasks, while services are provided by an Environment Manager. When a user moves from one environment to another, the representation of the task is moved, and service providers for the task are instantiated at the new location. Aura provides a

Context Observer to manage context. Context information is used to derive the intent of the user. The context observer merely collects context, and reports changes to the Task- and Environment Managers.

In Aura, depending on the detail of the collected context, the Context Observer might be able to derive the current task, location, and intent of the user. The intent is used for both tasks and location. If the user is working at home, but have a meeting scheduled at 10am and leaves his/her computer, the Context Observer might derive that the user is leaving for the office, and migrate the current task without intervention from the user. If the Context Observer is unable to derive the location or intention of the user because of insufficient sensors in the environment, the user must explicitly indicate this to Aura.

[72] proposes s a middleware for context-aware resource management, called CARMEN, in order to support the automatic reconfiguration of wireless Internet services in response to context changes without intervention on the service logic. CARMEN exploits proxies and the mobile agent technology to implement mobile middleware components that follow the provision-time movement of clients to support locally their customized service access. If a user moves to another environment the proxy will be migrated using wired connections. Each mobile user has a single proxy, which provides access to resources needed by the user.

When migrating, the proxy makes sure that resources are also available in the new environment. This done by: moving the resources with the agent, copying the resources, using remote references, or re-binding to new resources which provide similar services. The method is determined by inspecting the profile of the device. Each entity in CARMEN is described by a profile. User profiles contain information about preferences, security settings, subscribed services etc.

Service component profiles define the interface of services and Site profiles group together the profiles which all belong to a single location. Thus, context information in CARMEN describes the entities which make up the system.

CARISMA middleware is presented in [73] to deals with adaption of middleware depending on the needs of the applications. Profiles for each application are kept as meta-data of the middleware and consists of passive and active parts. The passive parts define actions the middleware should take when specific context events occurs, such as shutting down if battery is low. The active information defines relations between services used by the application and the policies that should be applied to deliver those services. The active part is thus only used when the application requests a service.

The CARISMA middleware plays auctioneer in an action protocol, where each application submits a single, sealed bit on each alternative profile. The auctioneer then selects the alternative which maximizes the sum of bids. To determine the bid each of the applications are willing to pay, functions which translate from profile requirements to values are defined. Like profiles, these functions may be changed at any time through reflection. This type of protocol makes sense because CARISMA delivers the same service to all participants.

The Cooltown [74, 75] is intended to support wireless, mobile devices to interact with a web-enabled environment. The basic principle is that devices, people, and things have a web-presence identified by a URL, which provides an interface to the entity. Users interact with the web-enabled environment using PDAs to interact with the available web-services. The PDA acts as an interface to the room, thus changing behavior based on location context. Other types of context might be used by web-applications by providing web-applications with other context like time or activity. The main principle in the collection of context is that it is provided by web clients.

The CORTEX project [76] project is concerned with research other than context-aware middleware, but has proposed a middleware to deal with autonomous mobile physical objects that cooperate with other objects, either mobile or static, by capturing information in real-time from sensors event messages.. CORTEX is based on sentient objects. A sentient object senses and views the behavior of neighboring objects, reason about it, and manipulates physical objects accordingly. Sentient objects dynamically discover each other, and share context information.

To support sentient objects, CORTEX provides a middleware based on component frameworks, each of which provides a service to the sentient objects: *Publish-Subscribe*, *Group Communication*, *Context*, and *QoS management*. Publish-Subscribe is used for discovery, while the other component frameworks support communication, context retrieval and inference, and arbitration of resource allocation. The resulting middleware is configured at deployment time and can be reconfigured at run-time through a reflective API to adapt to changes in the environment.

The Gaia Operating Systems [77, 78] is intended to be a *meta-operating system*. That is, a distributed middleware system providing functionality similar to an operating system. Gaia builds on the notion of an *active space*, coordinating heterogeneous devices in a physical space, typically a single room. Gaia manages resources throughout the active space. Gaia is structured like traditional file systems with a kernel providing the necessary services and applications build with an application framework on top.

Gaia differentiates between location, context, and events and although they can all be seen as different kinds of context, they are handled by different components. Context is collected by *context providers* and higher level context, such as activity, can be inferred from low level context. An additional presence service deals with which entities are present in an active space.

MiddleWhere [79] provides advanced location information to applications and incorporates a wide range of location sensing techniques in a model for location. Location information originates in *Location Providers* and is stored in a spatial database. A reasoning engine uses the location information from different providers to determine location and a location service uses the spatial database and the reasoning engine to provide location with a certain probability. The location model is hierarchical and deals with three different kinds of location: *points*, *lines*, and *polygons*. Each is represented by coordinates and a sym bolic name.

MiddleWhere deals with quality of the location information. The quality is measured according to *resolution*, *confidence*, and *freshness*. Resolution differs widely between different location sensing techniques. For example, a person using a card-reader to enter a room will tell the system that the person is somewhere inside the room while GPS has a resolution down to perhaps 10 meters. An RF badge might have a resolution of 1 meter.

Confidence is a measure of how precise the sensor is in terms of probability that the object is within the sensed area. This probability originates in the sensors which register the object and in the case of multiple sensors; the information is fused to yield a single value. Freshness is based on the time since the last sensor reading, and each type of sensor has an associated temporal degradation function which, based on freshness, degrades the confidence in the information.

MobiPADS [7980] is a middleware system for mobile environments. The principle entity is *Mobilets*, which are entities that provide a service, and which can be migrated between different MobiPADS environments. Each mobilet consists of a slave and a master. The slave resided on a server, while the master resides on a mobile device. Each pair cooperates to provide a specific service. Services are composed by chaining them together in specific order, and the slave mobilets on the server are nested in the same order. This provides reconfiguration based on different requirements.

MobiPADS is concerned with internal context of the mobile devices, which is used to adapt to changes in the computational environment. Adaption takes place in either the middleware based on system profiles, or by letting mobilets adapt to the events they receive. Based on the requirements in the profile, the service chains can be reconfigured to deal with a constrained environment, based on programmer provided alternatives service chains. Applications have access to reflective interfaces for context, service configuration, and adaption strategies, and can change them to obtain a different service from the middleware.

[81] proposes a solution to QoS specification, setup, and enforcement at the middleware layer. The idea is that the middleware cooperates with existing solutions at OS, network, and application levels. This solution spans from QoS specification and translation in the development phase of an application, to QoS setup and adaptation at runtime. During the development phase, the application developer specifies QoS parameters, possible configurations and applicable environments of an application.

The specifications are then translated by the QoS compiler, a companion development tool of the middleware, into internal representations, which will be injected into the middleware and used at runtime. During the runtime phase, the QoS-aware middleware performs QoS setup and adaptation for the application. QoS setup takes place right before the execution of the application, while QoS adaptation is triggered during the application execution by resource fluctuation, user mobility, and change of user preference.

EnTiMid, a middleware implementation developed in a house or building automation context is presented in [82]. The aim of this middleware is to offer a level-sufficient abstraction, making it possible for high level services to interact with physical devices (such as lamps, heater or temperature sensors). EnTiMid is a schizophrenic middleware [83] in which service access can be generated from an abstract services description. EnTiMid supports various services access models (several personalities): SOAP (Simple Object Access Protocol), UPnP [84] and DPWS (Device Profile for WebServices) [85].

EnTiMid is a layered middleware based on services and is structured around three layers: a low-level layer which manages communication with the physical devices, and provides an access to KNX, X2D and X10 devices; a middle layer (the kernel), offering a message oriented middleware (MOM), and giving a first abstraction level of low-level layer; a high-level layer which publishes services and enables the device access through standard protocol. EnTiMid defines a set of interfaces between the services provided by the low level layer and the services required by the middle layer to allow the access to the physical devices.

[86] presents the design and implementation of a middleware approach for context-awareness, and adopted fuzzy logic as an intelligent reasoning method for selecting data dissemination protocols in the design of the decision

mechanism. The approach uses a context space theory model shown in [87] for model fundamental nature of context and enable context and situation awareness for context processing. Its context model gives a common representation for context that all entities in the environment use of pervasive computing. Instead, it provides a common base on which various reasoning mechanisms can be specified to handle context.

This middleware contains separate physical, sensor platform, service, knowledge, context management, and application layers. Physical Layer: contains a variety of sensors and actuators which monitor and gather context information about the pervasive environment. A prototype of this middleware has been developed, which detect tourist' physical context and provide reminding.

4.6 Discussion

Several architectures of context-aware middleware have been proposed. We can distinct two approaches among proposed middlewares [57-61]: centralized or distributed. The most common design approach for distributed context-aware frameworks is a classical hierarchical infrastructure with one or many centralized components using a layered architecture.

This approach is useful to overcome memory and processor constraints of small mobile devices but provides one single point of failure and thereby lacks robustness. Most of these middlewares have a layered architecture. Generally, the common layers used in these architectures are:

- Physical sensing layer: witch collects contextual information.
- Data processing layer: it uses the raw information and deduces significant context information.
- Database layer: manages the raw and contextual information in the databases.
- Abstraction layer: hides the hardware heterogeneities and gives a uniform image of the network to the application layer. In some cases it provides programming and configuration possibilities to the system administrator who won't need technical information about the sensor and actuators installed in the network.

Architectural design of context-aware systems needs more efforts in order to provide an appropriate architecture that suits pervasive system requirements. Most common drawbacks in these architectures are:

- The proposed architectures are mostly specific to an application domain (localization systems, human-computer interaction, etc.) and require additional effort for their adaptation to other domains.
- Architectures based on a server suffer from the problem attributed to a centralized system: when the server breaks down, all other system components will be affected also it requires a dedicated hardware and software which increases its implementation cost. A centralized architecture contradicts the nature of contextual information in a pervasive computing system which is in general distributed and the mobility characteristic of devices in such environment.
- A pervasive system is characterized by its rapidly changing environment due to mobility; hence devices can be added or removed dynamically without affecting the entire operation of the global system which requires a dynamic and automatic devices and resources discovery mechanism. This aspect was not deeply discussed in most architecture and needs more attention in future systems.

Finally a number of existing approaches appear to be quite suitable and efficient but none of them appears to be a perfect match for the requirements.

Table VIII summarizes the capabilities of the surveyed solutions based on considered reference model and shows that comprehensive solutions do not yet exist. Our own middleware, which we discuss next, introduces decision support and addresses a large subset of the requirements listed in this table except security and privacy which are out of objective of this thesis.

TABLE VIII. SURVEYED MIDDLEWARES

Service or Layer / Middleware	Context abstraction	Supervision and QoS	Database	Context processor	Data management	Context router and aggregation	Context provider	Security and privacy
[61]	√	X	X	√	X	√	√	X
SOCAM	X	X	√	√	X	√	√	X
[64]	X	X	X	√	X	√	√	X
EMISS	X	X	X	√	X	√	√	X
PACE	√	X	X	√	X	√	√	√
ConStruct	√	X	X	√	X	√	√	X
Aura	√	X	X	√	X	√	√	X
CARMEN	√	X	X	√	√	√	√	X
CARISMA	√	√	X	√	X	√	√	√
Cooltown	X	X	X	√	X	√	√	X
CORTEX	√	√	X	√	X	√	√	X
Gaia	√	X	X	√	X	√	√	X
MiddleWhere	√	X	√	√	X	√	√	√
MobiPADS	X	X	√	√	X	√	√	X
[81]	√	√	X	√	X	√	√	X
EnTiMid	√	X	X	√	X	√	√	X
[85]	X	X	X	√	X	√	√	X

√ : OK X : KO

4.7 System architecture

Figure 27 shows the architecture of the system. The system has a multi-layer architecture. The Context provider, Middleware, Application and GUI layers are the main layers of the system.

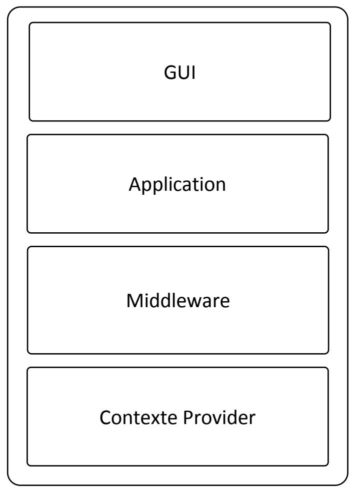


Figure 27. System architecture

Context provider layer and Middleware layer are responsible of aggregating the raw context and providing the deduced state of the context, each of these layers consists of different services, like: Context provider, Context aggregator, Context processor, Fuzzy Context Knowledge Base, Control Generator, Context abstraction layers. (Section 4.9 will give a detailed presentation about the middleware layer.)

Context provider is deployed in sensors to sense and transfer the raw context data into context aggregator layer of the middleware. It applies an abstraction to sensing data. Context supplier is a distributed application which is deployed in the sensor networks (Medical sensors, Home automation and other wireless sensor networks like ZigBee).

Application layer also consists of three services: Health, Activity monitoring and Health. This section explains each service and their functionalities. Context-aware Application uses the deduced context information provided by the middleware layer to do activity and health monitoring or to adapt the context rapidly changing situation of the user of other context (Comfort) for example changing the indoor light level by any change in outdoor light.

Graphical User Interface (GUI) layer provides the possibility for the healthcare providers and healthcare professionals to visualize the person's deduced health and physical states and also provides the possibility of controlling the sensor and actuators from distance, via Internet.

4.8 Data modeling

In CodaQ all data are given a uniform representation, and their level of abstraction is raised through the use of data format definitions for *raw event*, *deduced state, virtual sensor, zone, query and query reply* that will be detailed in the following.

4.8.1 Raw Event (RE)

The **Raw Event** is the output of a sensor on which no process was accomplished; it is the input of the application. The Raw Event shows an event detected by a sensor on a specific place, like movement or presence of a person, window or door opened or closed, light switched on or off. The Raw Event is presented by **RawEvent** parameter:

RawEvent (Id, Time, SensorID, {Values}, Zone)

For example **RawEvent**(10,22/11/2010-14:15:23,316,21,Room01), means that the raw event number 10 is detected by sensor number 316, at 14h15m23s of 22/11/2010, at Room01 and the measured value is 21.

4.8.2 VirtualSensor

The virtual sensor is the main component of hardware abstraction in our middleware. Virtual sensors abstract from implementation details of access to the sensor network and correspond either to data streams received directly from sensors or derived from other virtual sensors accessible through the network (thus a chain of virtual sensors).

A virtual sensor can be any kind of data producer and/or consumer, for example, a real sensor, a wireless camera, a desktop computer, a cell phone, or any combination of virtual sensors.

A virtual sensor may have any number of input streams and produces exactly one output data stream based on the input data streams.

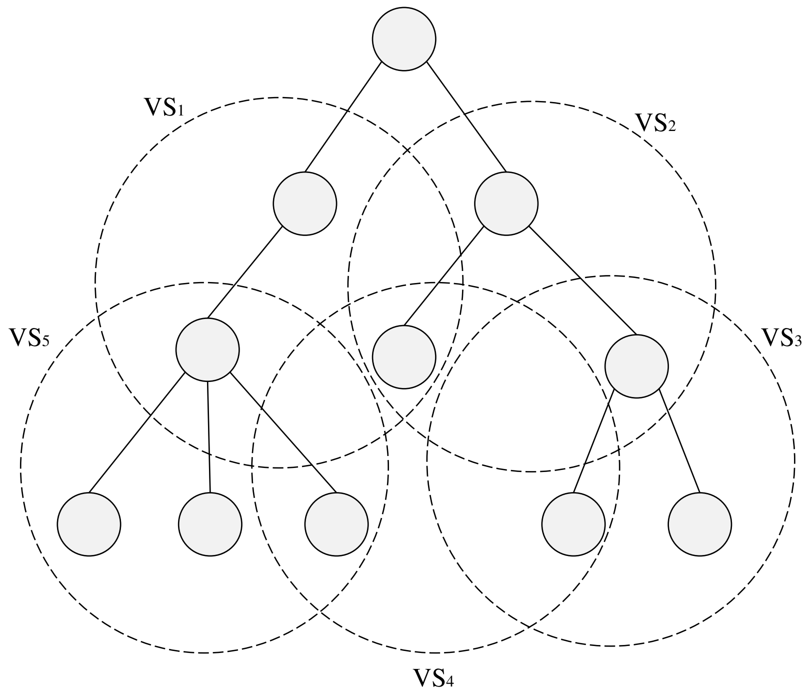


Figure 28. Virtual Sensor definition

The virtual sensor descriptor in our architecture is presented as:

VirtualSensor is a set of sensors (Id, ∑Sensors)

which provides all the necessary information required for deploying and using the virtual sensors, such as the data used for identification which is linked to the other information of the sensors, like their type, location, structure and properties of the produced data streams, a declarative XML-based specification of the data.

Figure 28 shows a sample clustered sensor network. In this figure, we can see 5 virtual sensors, VS_1...VS_5. As the figure shows, each sensor can be a part of many virtual sensors and each virtual sensor can be created by sensors in different clusters. Our middleware create these virtual sensors dynamically based on context state (state based virtual sensor formation). That means, for example in a telehomecarecare system for elderly, based on location of the elderly, the middleware will activate the formation of different virtual sensor.

4.8.3 Deduced State

The deduced state is the result of the processes in context processor on the raw events, and presents the context's actual situation. The deduced state is presented by **DeducedState** descriptor:

DeducedState (Id, Value, Category, Time, Zone/Place, VirtualSensorId)

We can say that a deduced state is out of a Virtual Sensor which presents the state of the context. For a user, the value of **DeducedState** could be: Fall, Leave/Arrive, Sleep, Eating, Reading, etc.

For example **DeducedState** (128,"Eating", "Normal", 22/11/2010-16:10:12, "Kitchen", 257), means the people was eating at 22/11/2010-16:10:12 in the kitchen, this state is a normal state and is deduced by virtual sensor which has Id 257.

4.8.4 Zone

Zone, presents the location of a given event or state:

Place/Zone (Id, ∑Sensors)

4.8.5 Query and Query Reply

Query is the command generated by the system to an actuator or a sensor. Light switch off, change the lighting level, open the door are some examples of a query. It could also be used for the supervision of the system in order to detect any failure in the hardware or the communication links. The query is represented by Query descriptor:

Query (Id, Category, {Sensors}, {Actuators}, Zone, Supervision (Bool)).

If supervision is true, then the answer will be a **QueryReply** which is presented by **QueryReply** descriptor as the acknowledgement of the sensors or actuators to a Query:

QueryReply (Id, QueryId, {Sensor}/{Actuator}),

and if supervision is false, the answer will be a **RawEvent**.

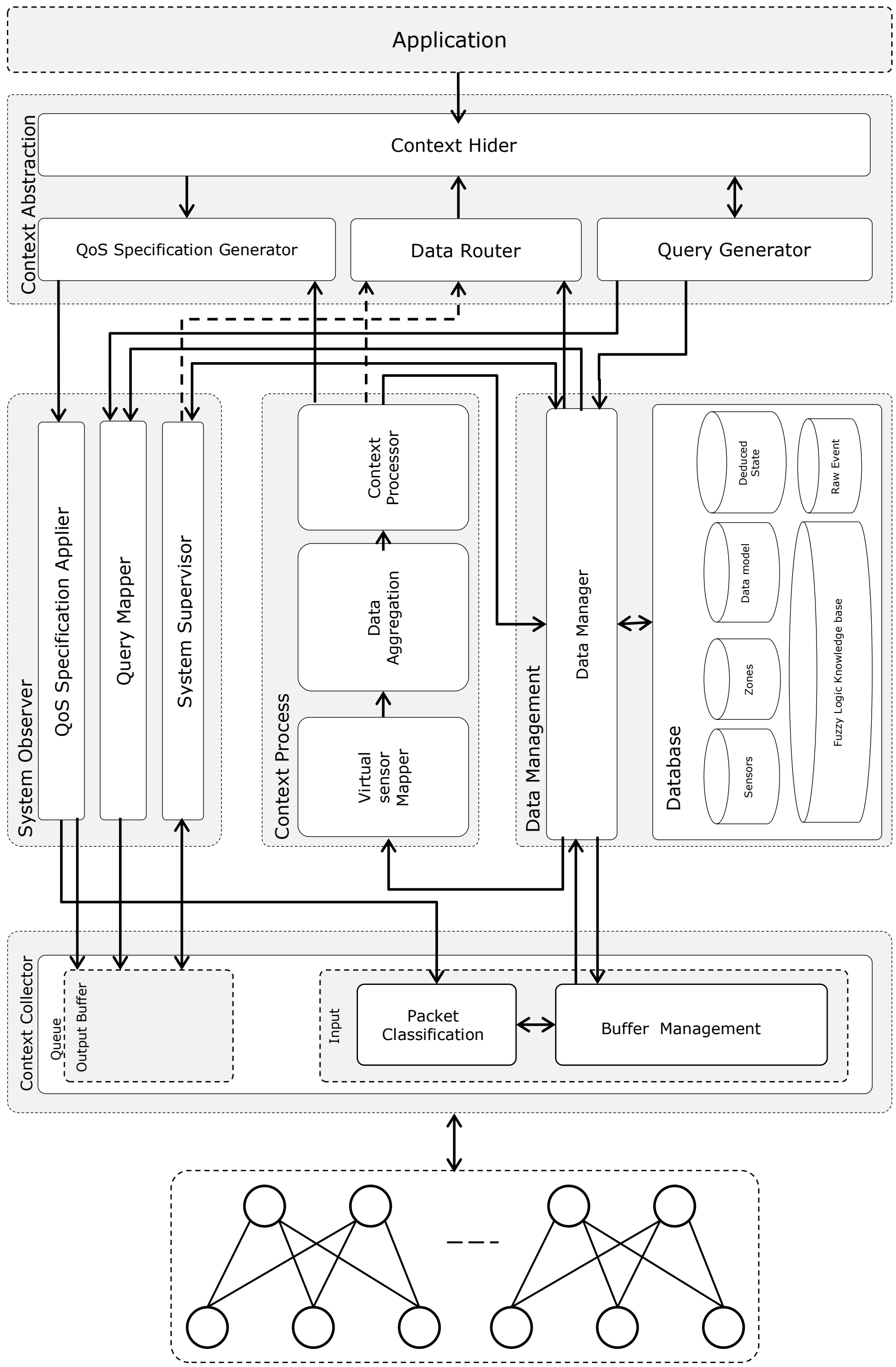


Figure 29. Middleware architecture

4.9 CodaQ – Context-aware Adaptive QoS Middleware

To ensure the context resolution, the system must be able to answer three main questions about each event: When? , Where? , What?

In order to guarantee this ability, our middleware layer is composed of different components (Figure 29). Next sections present the details of this architecture.

4.9.1 Context Collector

In a context-aware middleware, the context data collection is an important function. Context information is derived from an array of diverse information sources, such as location sensors, light and temperature sensors, computers and cameras.

In our architecture it is Context Collector that collects the context raw events from the distributed context providers and saves them in the Raw Event data base. This layer has a Queue Manager sub-layer which controls the input buffer by information received from QoS Specification Provided. For more information please see section 4.10.

4.9.2 Data management

This layer has 2 sub layers as figure 30 shows:

- **Data Manager:** receives queries from different components of the middleware, executes them and returns the results.
- **Database:** this layer stores the data from different parts of the middleware into 6 databases:
    - **Raw Event Database:** The Context Collector Layer receives the information about all events detected by sensors and stores the latest values of each sensor in raw event database.

This information is vital for context tracing. Context processor can perform deduced state of the context by using this information.

- **Deduced Context State Database:** all deduced context states' values are stored and managed in this database by context processor. Managing historical context data in this database provides the ability to implement intelligent learning algorithms which allow providing highly adaptable context-aware services. In addition, based on learning algorithms, contextual information can be predicted to proactively provide a certain set of services to the user.
- **Fuzzy Context Knowledge Base:** The context processor works by adopting fuzzy rule based techniques. Fuzzy rule based approach deduces which is the state of the context in a specific situation by using a set of fuzzy rules (IF-THEN rules). Table IX shows some example of the rules. Such rules can be specified by healthcare experts or a particular healthcare giver. They can also be obtained by using learning techniques from user's behavior or activity pattern which is beyond of the scope of this PhD work but may be one of interesting future works.

TABLE IX. SOME EXAMPLE OF FUZZY RULES

IF *Temperature* is *High* THEN Send *Low* to *Heater*
IF *Blood Sugar* is *High* and *Time* is *Night* THEN Send *High Sugar Alert*

- **Data model:** it is the data model definition database presented in 4.8. Query Generator and QoS Specification

Generator use this database to applying the data model to the Query and QoS Specification. By applying the data model definitions the middleware hides the heterogeneity of the network in order to guarantee the interoperability between them.

- **Zones:** manages the location information of the context.
- **Sensors:** all necessary information about all sensors and actuators installed in the networks are saved in this database.

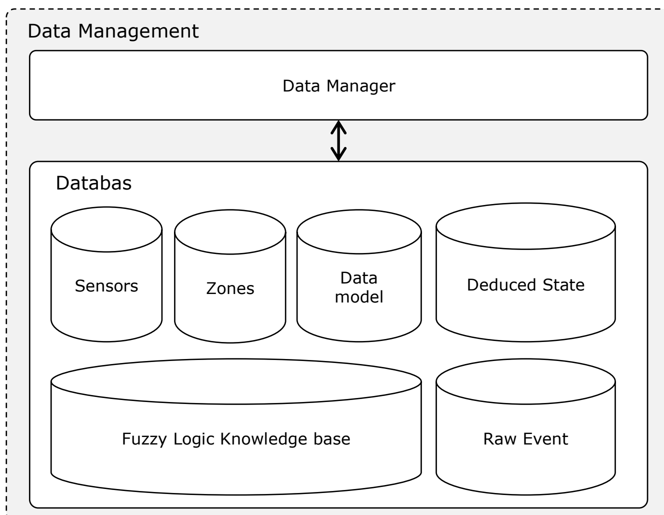


Figure 30. Data Management

4.9.3 Context process

Deduces real context state from raw events and states (basic sensed context data) using fuzzy rule-based reasoning techniques, and also checks for knowledge consistency in the fuzzy context knowledge base. This layer has 3 components, figure 31 shows these layers:

- Data Aggregator: aggregates the context information by using Raw Event Database and Virtual Sensor information and provides them to Context Processor Layer.
- Virtual Sensor Mapper: provide necessary information about all virtual sensors of the network. It uses Sensor and Zone Databases to create virtual sensors and provides them to the Data Aggregation layer.
- Context Processor: uses information provided by Data Aggregation layer, processes them in order to deduce real state of the context and saves deduced states in the Deduced State Database.

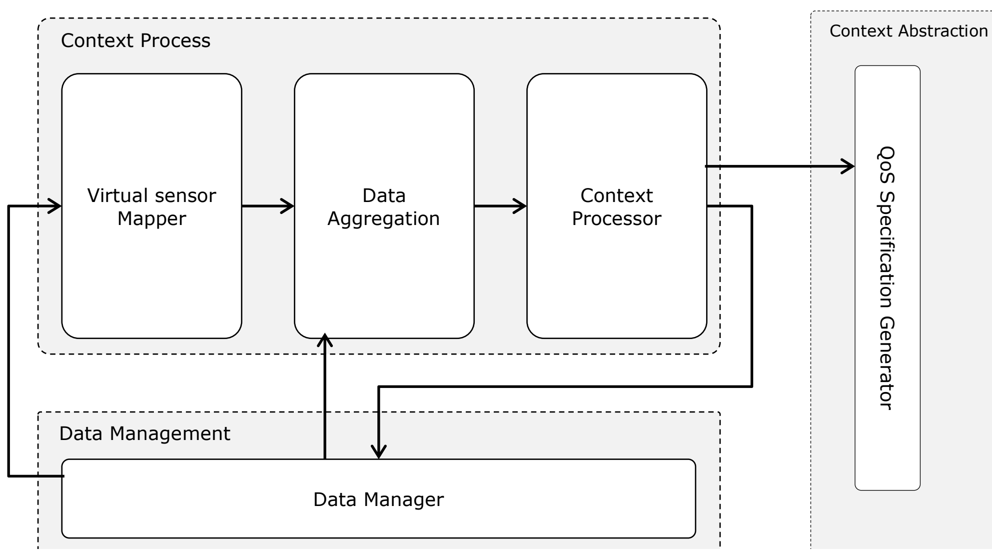


Figure 31. Context Process

4.9.4 System Observer

QoS observer uses run-time information about the environment to choose the most suitable configuration of applications and middleware services. This requires that the middleware has a specification of elements in the environment, and QoS specifications.

As shows figure 32, this component has 3 layers:

- **QoS Specification Applier:** uses QoS specifications from QoS Specification Generator and sends them to the context router layer. For more information about QoS managing in our system, please see section 4.10.
- **System Supervisor:** it creates Queries and sends them to the specified destinations (Sensors and Actuators) in order to monitor health state of the installed sensors and actuators and detect any failure in them or the communication links.
- **Query Mapper:** uses Sensors and Zones data base and generate the necessary queries in order to provide the latest and fresh information of the sensors.

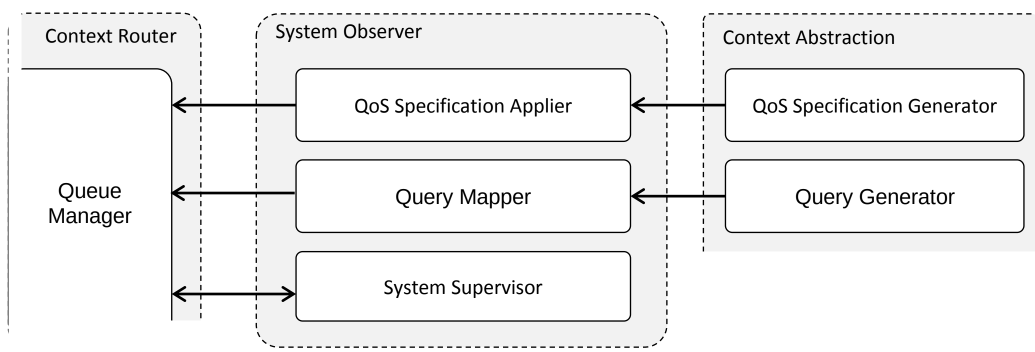


Figure 32. System Observer

4.9.5 Context abstraction

In a home domain various devices exist, for example sensors, medical sensors, embedded sensors, traditional computing devices, and communication devices. These devices have different platforms, and a context-awareness middleware can collect context information from these heterogeneous devices.

Since in our context-aware middleware all elements are services which highly abstract the middleware operations, and provide consistent interfaces and uniform protocol with standardization manners, therefore we overcome heterogeneity in ubiquitous computing environment.

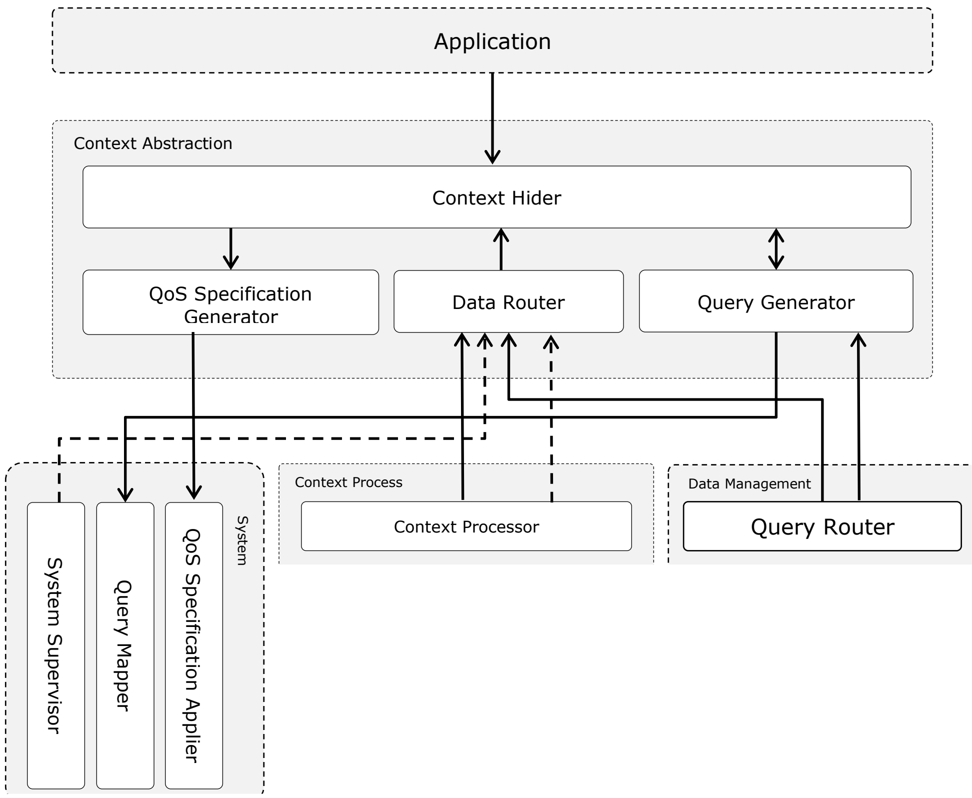


Figure 33. Context abstraction

This layer uniforms the information and data transferring between Application and the Low level layers (Middleware and Context providers) by applying ontologies. It has 3 layers (Figure 33):

- **Context Hider:** receives the requests from application and sends it to QoS Specification Generator and Query Generator layers. It also receives results of the queries and sends them to application layer.

- **Query generator:** generates persistent controls on demand of application layer. This layer is bridge of communication of application layer with the sensors and actuators and allows applications to extract desired context information from the networks.
- **QoS Specification Generator:** generates QoS Specifier by using QoS need of the application, received from Context Router layer.

4.10 Context-based Adaptive QoS

As discussed, in our remote healthcare system, different types of sensors (Medical sensor, Home automation sensors) are used to provide information about the context, that means we have different networks and the data have different characteristics which demand different QoS requirements (Delay, Packet loss).

The QoS offered for these networks depends on the actual state of the context (like user's actual location, time …) and also on available communication network at a certain location and time and the performance of these networks. The offered QoS of these networks can change over time because of user mobility, changing number of users, available energy resource of a given sensor at a given location, etc. Therefore, we denote this information as context-based QoS.

The adaptive QoS service should have the run-time context information. This will guarantee that a context change will result in a change of the delivered QoS. That means the QoS method would always select a communication network infrastructure that offers the best QoS requirements. In contrast, in a context-unaware case, the system uses some default communication network.

Therefore, the choice for a communication network is a design-time choice. If we use context-based QoS, the selection of a communication network could be adapted according to the context changes and end-user's QoS requirements. In our design QoS observer applies QoS in 2 levels: Embedded and State-based QoS.

4.10.1 Embedded QoS

Because of mobility of the user, the underlying ZigBee networks, and the QoS offered by these networks, can change as the user moves therefore the sensors of the system are implemented by our adaptive QoS and dependability algorithm to provide QoS.

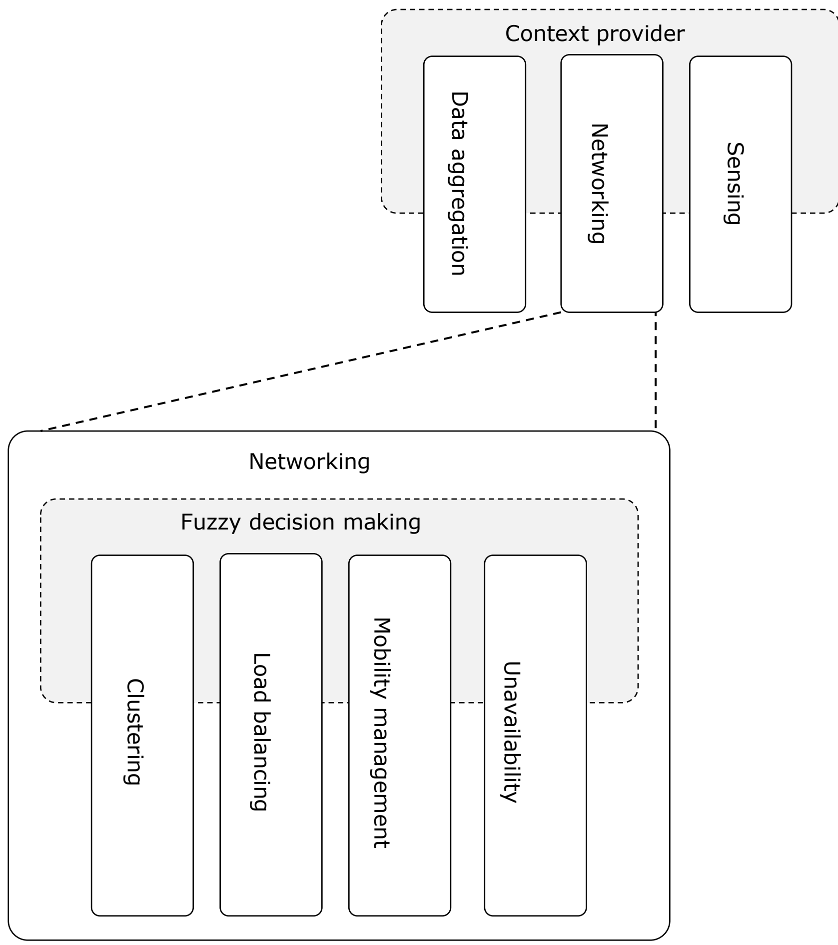


Figure 34. Networking Layer of Context provider component

Note that, in our system this only applies to the WSN part and we assume there is sufficient communication resource in the other networks. We use FLoR routing algorithm which optimizes the resource utilization by considering

Energy, *Mobility*, *Unavalibility history* and *Load* of the nodes. Clusters are also formed dynamically to adapt to the network changes. Our node admission policy described in chapter 3 is based on the routing feasibility of the system under this algorithm. (The detailed discussion of the routing algorithm can be found in Chapter 3).

By applying these solutions into the system, we will have three parts that manage the QoS (Figure 34):

- QoS and load balancing via sensor admission control
- Mobility management
- Unavailability management

QoS management service is used to select the correct promotion service considering the battery level and the available memory.

4.10.2 Run-Time State-based QoS

The QoS requirements of the sensor data are determined by sensor data type and condition of the patient. For example, the medical information like ECG, blood oxygen levels may require a higher level of reliability than video sensor information, or when a patient is having an unusual heart rate, the ECG data are highly vital and require higher reliability and lower delay.

QoS Specification Provider uses run-time information about the context to choose the most suitable configuration of applications and middleware services. This requires that the middleware has a specification of elements in the context, and QoS specifications. Some context information that can be used to improve the QoS are:

- Information about the users, such as location.

- Information about system requirements in a given mission, the roles of each sensor and type of information in them, and the importance of data delivered by them.
- Information about the resources, devices, and connections in the system, such as memory capacity, variations in communications, and power.

With this context information, the Context Collector can prioritize and order the information delivery based on which information elements are most important to the user, and can select information processing based on user characteristics and resources, and allocate resources to the most important operations.

We use XML based specification to describe QoS requirements. However, in order for a client application to be able to operate effectively with a given cardinality constraint, it needs to be able to determine how sensor data is mapped onto defined data model. The system observer layer does this. It provides an infrastructure to allow applications to describe their structure and query patterns which are generally referred to as QoS specification.

4.10.2.1 Queue Management

For now, our QoS management policy is based on priorities, defined by application QoS Specification Applier. The Context Collector has 2 principal components to manage the buffer queue (Figure 35):

- **Packet Classifier:** it receives data from the network. It stamps the packets by local time and uses QoS Specification to gives priority to the packets from specified sensor. And then sends the packets to Fuzzy Buffer Manager.

- **Fuzzy Queue Manager:** this component presents a fuzzy based queue management that performs buffer allocation and packet dropping dynamically and continuously by using Fussy Knowledge base and the priority added to the packets by Packet Classifier. The Fuzzy Queue Manager considers stamped packets in the buffers.

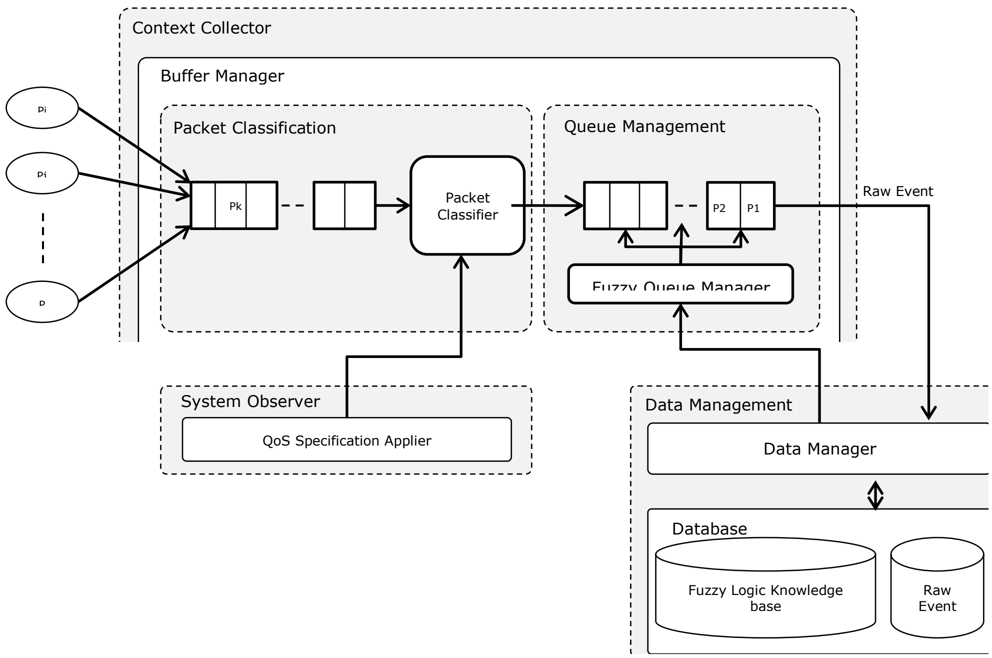


Figure 35. ueue Management in output buffer

4.10.2.2 Fuzzy Queue Management Algorithm

Table X shows the fuzzy based queue management. The buffer manager utilizes maximum available buffer, priority, packet size, flow rate, in a fuzzy controller to compute Urgency parameter of the packets in order to perform buffer allocation and packet dropping.

There are three steps involved in the controller such as fuzzification, inference with rule base and defuzzification. In the fuzzification step, crisp inputs are converted into linguistic values (such as low, high or medium), each of which is represented by a fuzzy set. Each fuzzy set is associated with a membership function used to characterize how certain the crisp input belongs to the set. For a given crisp input, the membership function returns a real number between 0 and 1. The closer the membership value is to 1, the more certain the input belongs to the set.

TABLE X. FUZZY BASED BUFFER MANAGEMENT ALGORITHM

```
Receive (Packet P)
{
  If (buffer is available)
    Send (P);
  If (buffer is not available)
  {
    FindMembership (Priority, Packet Size, …)
    Compute Urgency of P    /*by using inference, rule base and defuzzification*/
    If (P.Urgency < smallest Urgency of the buffer's packets)
      Drop (P)
    Else
    {
      Drop (packet with smallest Urgency)
      AllocateInBuffer (P)
    }/*else
  }/*if
}/* Receive (Packet P)
```

A single crisp value can take more than one linguistic value if the membership values overlap. In the inference step, a set of rules called rule-base, which emulates the decision-making process of a human expert is applied to the linguistic values of the inputs to infer the output fuzzy sets which represents the actual control signal for the queue management.

The output fuzzy sets are defuzzified to generate the crisp values of allocation factor and dropping factor using centroid defuzzification method. The computed Urgency parameter is informed to buffer manager for allocation decision. Whenever allocation is not possible by buffer manager, packet dropping is done by Urgency parameter.

4.10.3 Spatial and Temporal Consistency

In addition to the adaptive QoS, spatio-Temporal consistency is also very important to consider in a remote sensing system. Temporal validity is defined as the period of time to which a single instance of context information is applicable, and spatial resolution is defined as the precision with which the physical area, to which an instance of context information is applicable, is expressed [88].

Time is important for classification of situations of a given state and event. It is also highly relevant for sensor data that is acquired from a distributed environment. Using concurrency or exploiting the fact that events take place coincidental or within close timely boundaries are a basic way of relating different aspects of complex situations. Actions and events do not occur in isolation, and things that seem odd as individual actions take on different interpretations once fleshed out with further information. Spatial and temporal reasoning means reasoning in a similar way to that in which a person would deal with time and space in everyday life, and not the way how a physician, for example, would describe locations, regions, time points, intervals, events etc. and reason about them. [88]

Regarding the time aspects when acquiring sensor data and contexts a similar semantic as for location is observable. Values are created at a certain point in time; and these values are in general more relevant to an event that happens roughly at the same time than to an event that takes place much later or earlier. The concept of time should also be included in the model, exploiting the following basic observations:

- context information has a time of origin
- the relevance is maximal at the time of origin
- relevance decreases with an increasing time distance from time of origin
- if several sensors of the same type are available the one which provides the most timely reading has the highest relevance

In certain application areas it may be useful and beneficial to relate the relevance to issues that are specific to the application rather than to the temporal and spatial distance.

In context-aware activity monitoring systems, we are not usually interested in precise descriptions of space and time. A vague representation suffices to deal with the problems that we want to solve. For example, the output of Cooking Virtual Sensor will be: "The person is cooking", with information about the exact coordinates for this activity. The middleware will add the other information about place and time: "The person is cooking and at lunchtime in kitchen". This presentation is enough to decide whether the behavior is normal or not.

In order to guarantee spatial consistency, CodaQ uses the concept of the virtual sensor. The sensor network is the set of sensors installed in a same place or zone. That means we cannot use the light level of the kitchen and the temperature of the bedroom to form a virtual sensor and detect an event at a given place. But in the application level we can use the information with no temporal and spatial consistency; for example we can use the blood sugar level of a patient in 13 o'clock with the WC's presence sensor in the 13:30 to deduce a state. In this example we have neither temporal consistency nor spatial consistency.

4.11 Discussion and concluding remarks

The experience gained through our research and development activities showed us the lack of a bridge between the healthcare applications and the sensor networks in a telehomecare system. A bridge:

- to guarantee QoS needs
- for supervision of the system
- to deduce real state of the user and his environment
- for configuration and installation of the sensor and actuators
- to hide the heterogeneity and complexity of the system and sub systems

For these reasons, we developed a context-aware and QoS-aware middleware, CodaQ, which has an innovative architecture. Context Collector, Data management, Context process, System observer, Context abstraction are the main components of the architecture of CodaQ. In this middleware all data are given a uniform representation, and their level of abstraction is raised through the use of basic data modeling. A context-based adaptive QoS method was implemented on the system, in 2 levels: Embedded and State-based QoS.

CodaQ provides key features for both context management and their distributed environment installed in the sensors. It is decomposed into a number of components, offering certain functionality for applications.

CodaQ is oriented to provide functionality for context-aware applications. It has a common interface for context sensors, so it is easy to embed a new sensor to the system. The sensors can reside on physically distributed devices. Hence, the context information is forwarded to a centralized context component that performs further context delivery and processing.

Besides, the context component is responsible for context recognition. These context-related operations are performed at middleware layer, which makes context processing and recognition transparent for applications. The context-based storage is the component that provides data storing facilities.

QoS specification and supervision of the system are the other responsibilities of the middleware. In this case, this is the system observer that supervises the system and sends the QoS specifications to the system.

CodaQ is developed to work in the mobile environment and this is under responsibility of context provider component which is installed in the nodes. However some architectural design points are always open to discuss.

First point is that in this architecture, we gave the responsibility of context deduction to the middleware. That means the middleware does a pre-process to the raw data and sends the state of the context to the application layer. We do this to reduce application complexity and provide it the context information. However, we can do it otherwise by giving context processing task to the application. In this case the architecture will be presented as shown in the Figure 36.

The second point is that deploying the wireless sensor network on a smart home introduces the additional challenge of protecting information among users, which produces the need for robust security and privacy management. These aspects should be considered in the architecture of the middleware; however these are out of subject of this thesis. Refer to [89], [90] and [91] for more information.

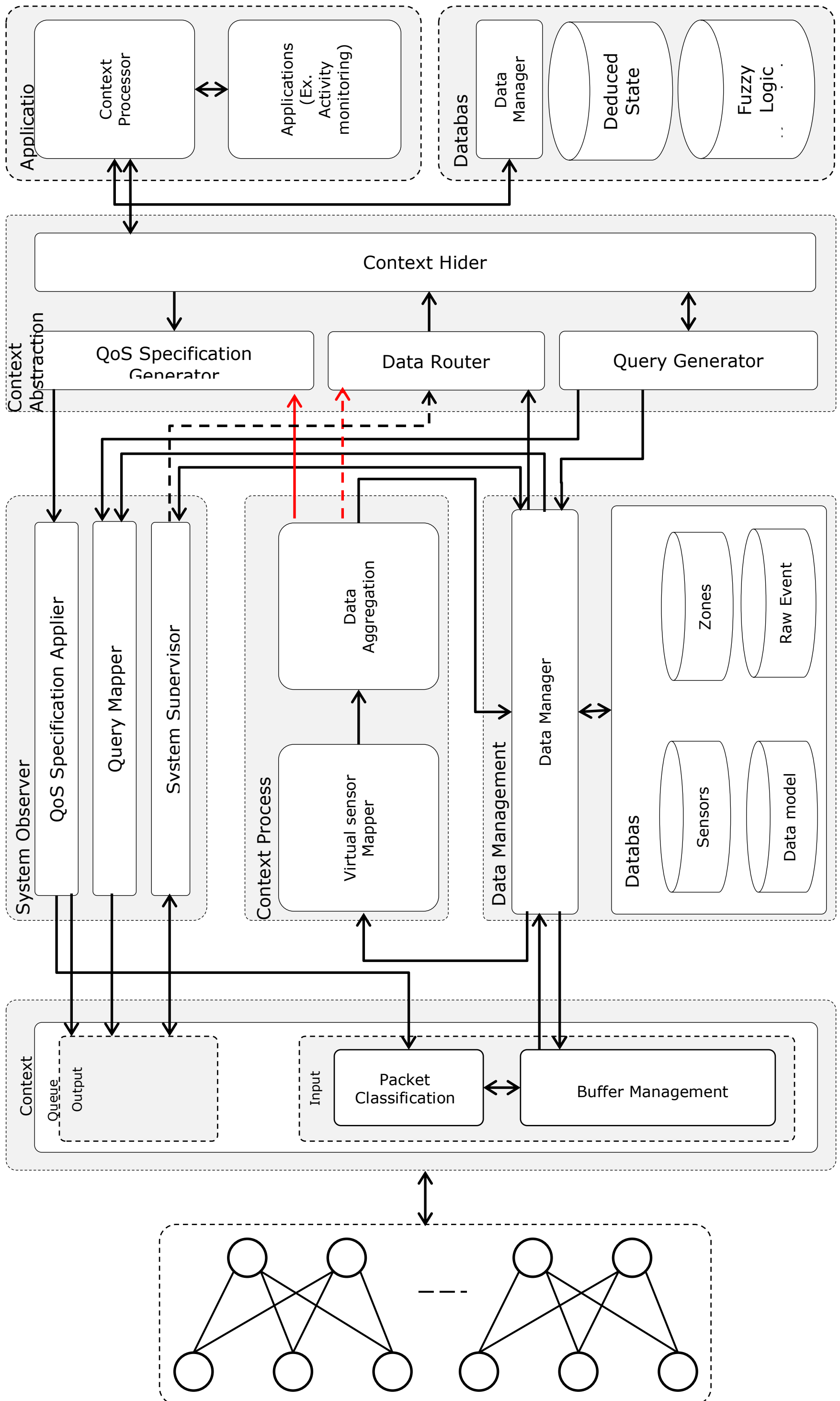


Figure 36. Modified architecture

Chapter 5 Implementation

This chapter describes the implementation and development approach that was followed to evaluate our context-aware system and the middleware architecture supporting it.

5.1 Introduction

System validation can be verified by showing its feasibility by developing a prototype which will demonstrate that the concept and its components are correct. However, to perform any meaningful validation, we require a real world test of the system, in semi-real environment. A prototype has been implemented to verify the feasibility of our proposals, based on our laboratory apartment in Colmar.

The programming environment of the middleware is Visual Studio .Net 2010 C# and ASP .Net web services and the service discovery is based on SOAP. The prototype utilizes a SQL server database. All the communication between components of the middleware and also the QoS specifications are performed by XML library. The prototype utilizes Imote2 sensors in the ZigBee network side. The programming environment of this part is Imote2.Biulder SDK under Visual Studio .Net 2005 C# and Microsoft .Net Micro Framework 2.5, based on TinyOS 2.0.

5.2 General view

Figure 37, shows the general concept of the prototype. In the client side (Home network), the system proposes many different services:

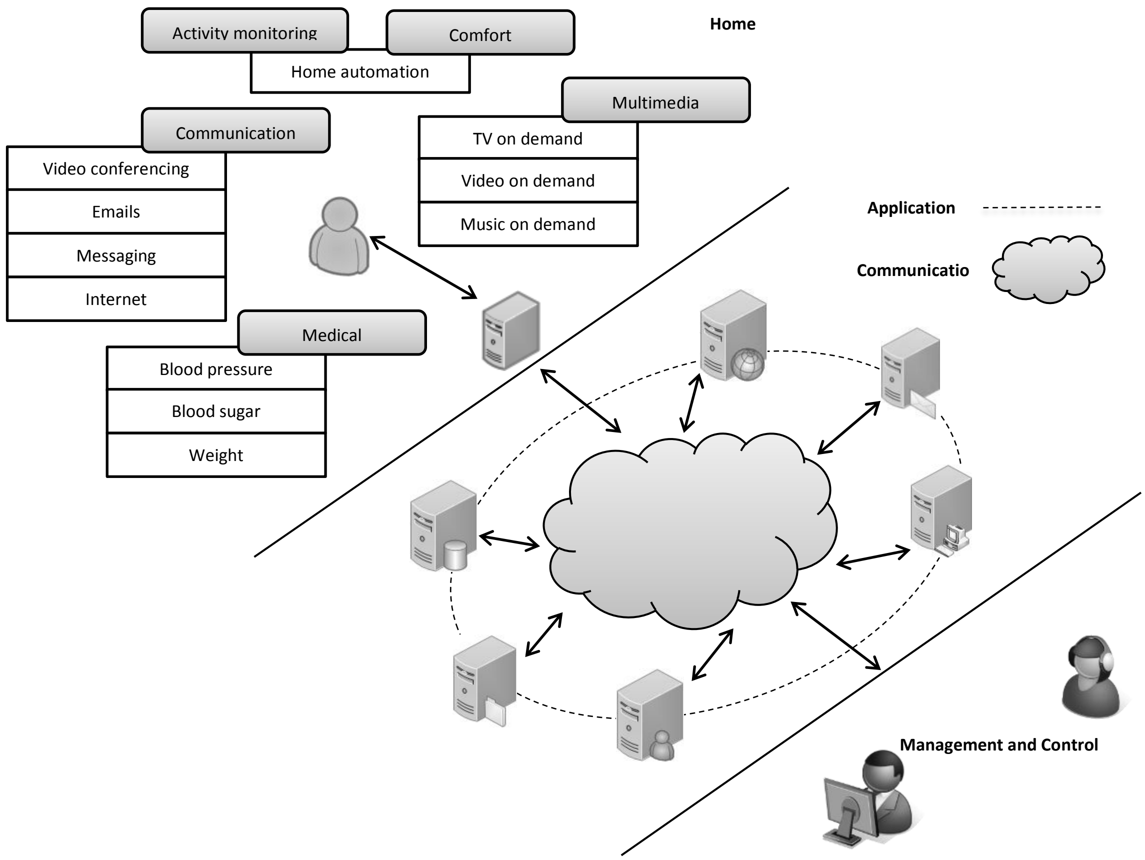


Figure 37. General view of the prototype

- Activity monitoring and Comfort: by using Home automation sensor and actuators. The system uses a more passive method. It senses person's presence in a room by home automation sensors installed throughout the living areas, such as motion and presence detectors, light sensors.
- Medical: to monitor the Blood pressure, Blood sugar and Weight, by using Bluetooth medical devices,

- Communication: Video conferencing (with family or with a doctor), Internet, emails and instant messaging,
- Multimedia: TV, Video and music on demand.

The home gateway, a PC, is the responsible to connect the home network with the server platform (database server, web server, application server…) via Internet. The system proposes also a management and control platform for remote access to the server platform or the home network.

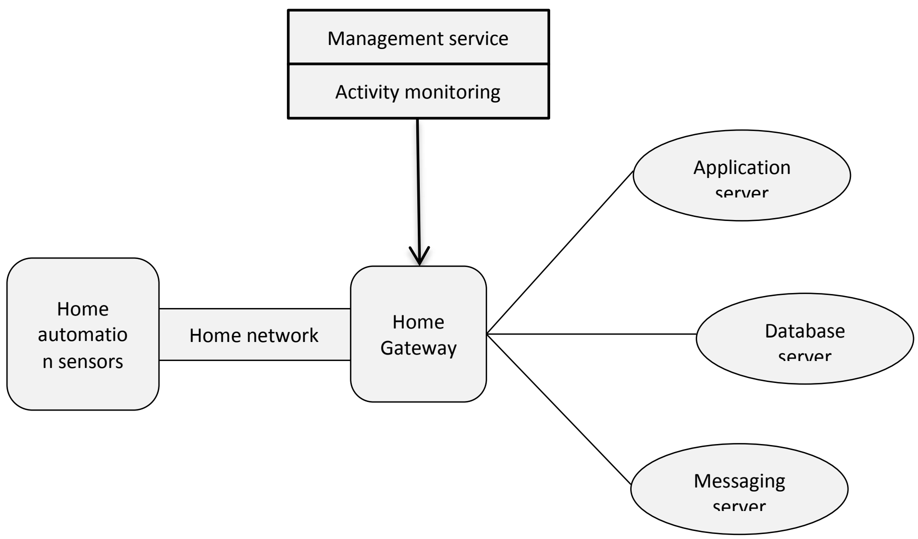


Figure 38. Activity monitoring and management service in Home gateway

The home gateway communicates via the home network with the home automation sensor and actuator networks (Figure 38). An application in the home gateway is responsible of filtering the data received or collected from the home automation sensors. For example if the person moves in a room, the motion detector installed in the room will be activated during this movement and the gateway will receive a lot of information about this movement which are not suitable for activity level estimation. The information that we need are the place of the person (kitchen, WC, bedroom…), the time that he will rest in this place, the frequency of going in and out to this place (this frequency will be calculated in the application server). In order to process the information, in one hand, the daily or weekly, activity patterns are analyzed in the central server for unusual behavior of a discrete illness, by using intelligent method based on Fuzzy Logic. On the other hand, if the gateway detects any inactivity in a specific place, for certain time, it will generate an urgent local alert to a medical call center.

The application installed in the home gateway, has a management part also. This part is the responsible to detect any failure or crash in the sensor and actuators, in order to guaranty the right information about activities of the person and also to reduce the number of negative positive alerts.

5.3 XML-based data presentation

5.3.1 Sensor installation and configuration

The system adds the installed sensors on a XML file, named Installation.XML. (Table XI). System Configuration and Installation XML presentation

TABLE XI. INSTALLED SENSORS


```xml
<?xml version="1.0" encoding="utf-8"?>
<Installation>
  <Location Name="Bedroom">
    <Sensors>
      <Sensor>
        <SensorType>Temperature</SensorType>
        <SensorNetworkId>101</SensorNetworkId>
        <SensorX>108</SensorX>
        <SensorY>85</SensorY>
      </Sensor>
      <Sensor>
        <SensorType>Light</SensorType>
        <SensorNetworkId>201</SensorNetworkId>
        <SensorX>108</SensorX>
        <SensorY>85</SensorY>
      </Sensor>
  </Sensors>

    </Location>
  <Location Name="Kitchen">
    <Sensors>
      <Sensor>
        <SensorType>Temperature</SensorType>
        <SensorNetworkId>0</SensorNetworkId>
        <SensorX>92</SensorX>
        <SensorY>253</SensorY>
      </Sensor>
    </Sensors>
    </Location>
<Location Name="WC">
…………………
```

This file uses the data model defined in the chapter 3. As table XI show, in the implementation of the CodaQ, we adapted the data model to able the application to present the graphical aspects of the interface. Here we can see that for each sensor we save its geographical coordinates, (X, Y). When the administrator exits the application and restart it, the application will be able to load and graphically present all the installed sensors on the plan.

5.3.2 Raw event

Table XII shows the RawEvent.XML file which presents the values detected by the sensors. By receiving the data, the system stamps the data and saves them in the data base. The raw value, location the network Id of the node and the time are the fields saved for each event.

TABLE XII. RAW EVENT XML PRESENTATION

```
<?xml version="1.0" encoding="utf-8"?>
<RawEvents>
  <RawEvent>
    <Id>1</Id>
    <Time>11/11/2010 22:57:27</Time>
    <SensorNetworkId>101</SensorNetworkId>
    <Value>1291</Value>
    <Location>Bedroom</Location>
  </RawEvent>
  <RawEvent>
    <Id>2</Id>
    <Time>11/11/2010 22:02:11</Time>
    <SensorNetworkId>201</SensorNetworkId>
    <Value>120</Value>
    <Location>Kitchen</Location>
  </RawEvent>
    ....................
```

5.3.3 Context information providing – deduced state

The context processor provides the deduced states by processing the raw events and saves them in the data base. The deduced states are transferred between the components of the CodaQ in form of xml files. Table XIII shows an example of representation of deduced events in CodaQ.

TABLE XIII. DEDUCED STATE XML PRESENTATION


```xml
<?xml version="1.0" encoding="utf-8" ?>
<DeducedStates>
  <DeducedState>
    <Id>1</Id>
    <Value>Eating</Value>
    <Category>Normal</Category>
    <Time>11/11/2010 20:10:31</Time>
    <Location>Kitchen</Location>
    <VirtualSensorId>801</VirtualSensorId>
  </DeducedState>
  <DeducedState>
    <Id>2</Id>
    <Value>Sleeping</Value>
    <Category>Normal</Category>
    <Time>11/11/2010 23:10:31</Time>
    <Location>Bedroom</Location>
    <VirtualSensorId>804</VirtualSensorId>
  </DeducedState>
  <DeducedState>
    <Id>3</Id>
    ....................
```

5.4 Sub-systems of the prototype

5.4.1 CodaQ Middleware

CodaQ was partially implemented in the prototype. Context collector, Data base, Data management, Context processor and Abstraction layers are the implemented components of CodaQ.

5.4.2 Application layer

Application layer of the prototype consists of Installation and supervision, User and Healthcare provider applications.

5.4.2.1 Installation and supervision system

Figure 39 shows the administration system interface of the system. The main form shows the plan of the apartment and its different rooms.

Figure 39. Installation and Configuration Interface

Note that here the technology behind each sensor is hidden to the user. That means he doesn't need to have technical information about the technologies.

Figure 40 show the installation interface of the system. In our system, adding the sensors will be done by a simple right clicking and choosing the sensor type in the context menu appeared on the interface.

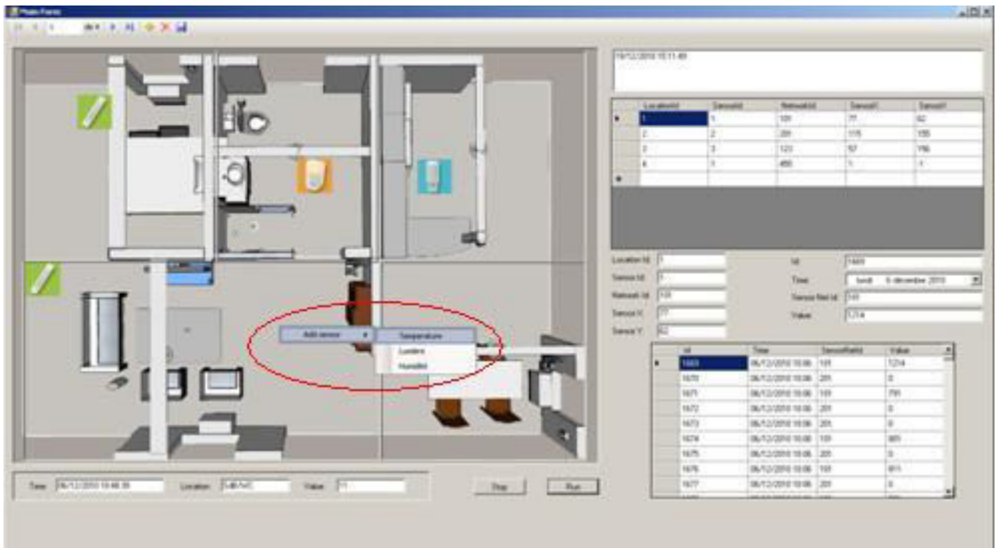


Figure 40. Installation and Configuration Interface – Add sensor

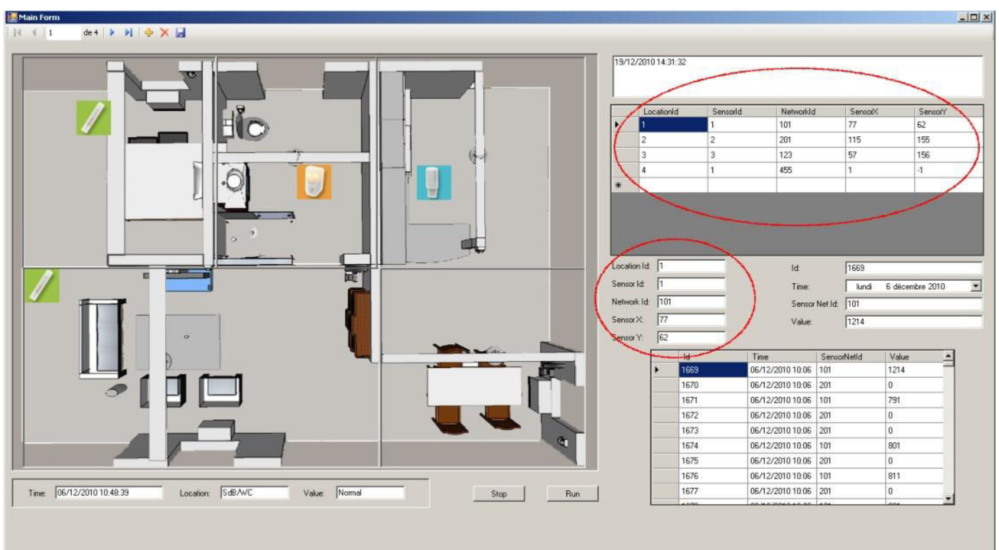


Figure 41. Visualizing installed sensors and their details

In the right side of the interface we can see the list of the all sensors installed in the system with the detailed information about the sensors and also the data collected from the sensors (Figure 41).

After installation, the administrator will be able to supervise the values detected by each sensor. It must just place the mouse pointer on the sensor to load and see the value (Figure 42).

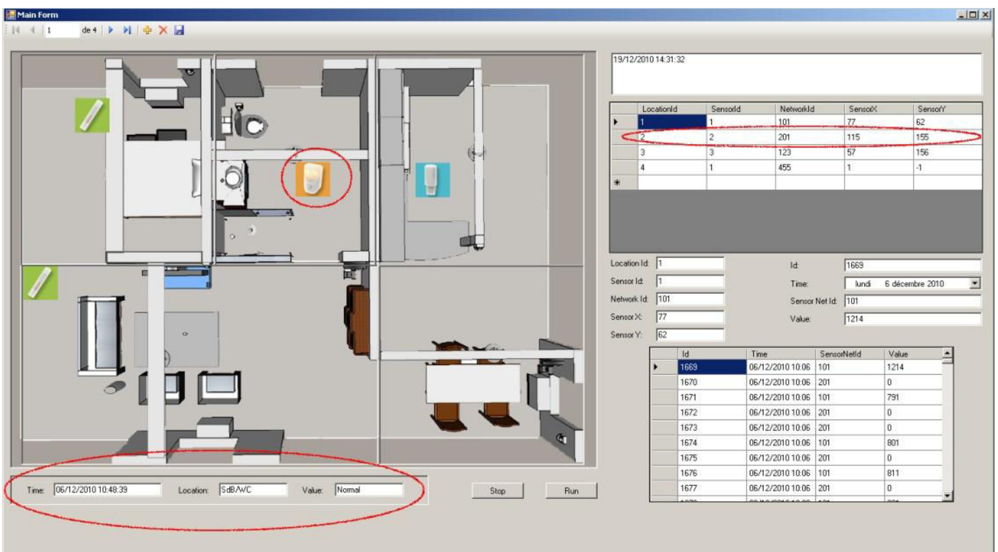


Figure 42. Supervision of the sensor value

5.4.2.2 User side

The system gives to the users a full control of Home automation devices, by using system's simplified interface. The prototype is integrated with a simplified and user-friendly videoconference system which is adapted to elderly needs. Telecare is related to offer health care services over Internet or a wide area network, and in this domain, videoconferencing is one of the important solutions, allowing real-time interaction between patients and the Doctors. The offered system is integrated with a simplified and user-friendly videoconference system which is adapted to use in elderly telecare. The system enables the user to have a video meeting by the families and also to request an online consulting; in this case the procedure is described in figure 43.

In the client part, Video and Music on demand, TV, Photo album and calendar are the rest of the prototype, but these parts are beyond the objective of this thesis, but they are integrated to the system to present a complete prototype demonstration. Figure 44 and 45 show 2 pages user interface. In the first one, we can see the home automation control and supervision interface and on the other one, the blood pressure page of the health interface.

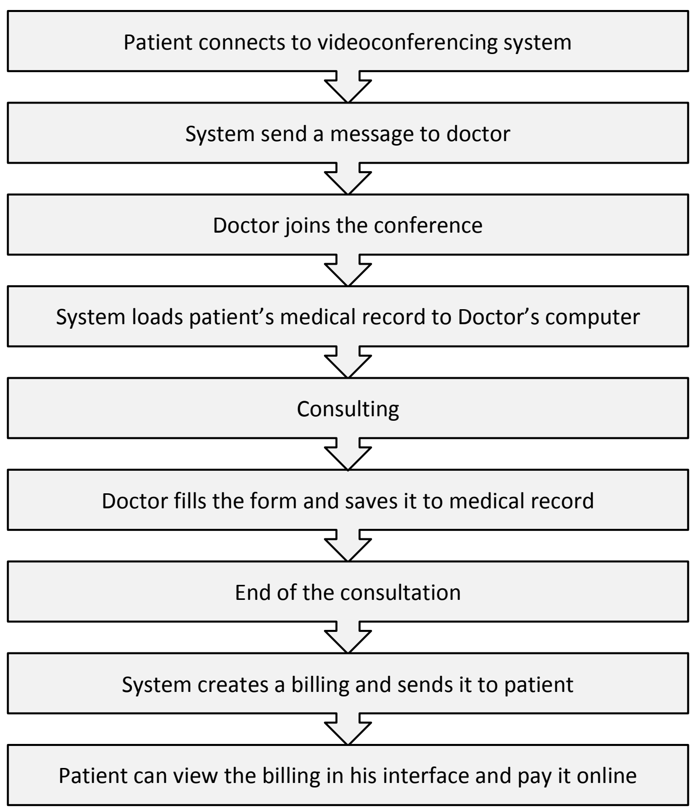


Figure 43. Teleconsulting procedure

Figure 44. Home automation page – User interface

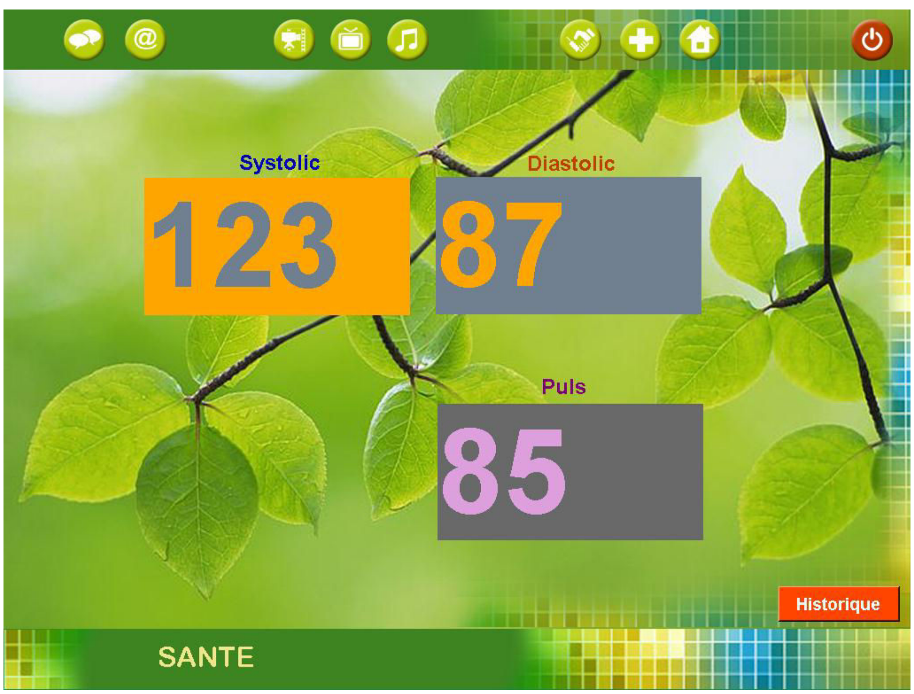


Figure 45. Health page – User interface - Blood pressure results

5.4.2.3 Healthcare provider side

Healthcare providers are interested by deduced state of the context. Therefore the system must be able to provide a complete and comprehensive presentation of the deduced states. Figure 46 shows an example of health giver's interface. In this figure the system generated a blood pressure report of a patient for one week.

Figure 46. Blood pressure report interface

5.4.3 Sensor nodes side

In the sensor side, context collector and FLoR routing have developed. All these modules are implemented by Microsoft Imote builder SDK in visual studio 2005.

Context collector consists of a TinyOS application which collects the luminosity of the environment and sends it to the base station even 10 seconds (Figure 47).

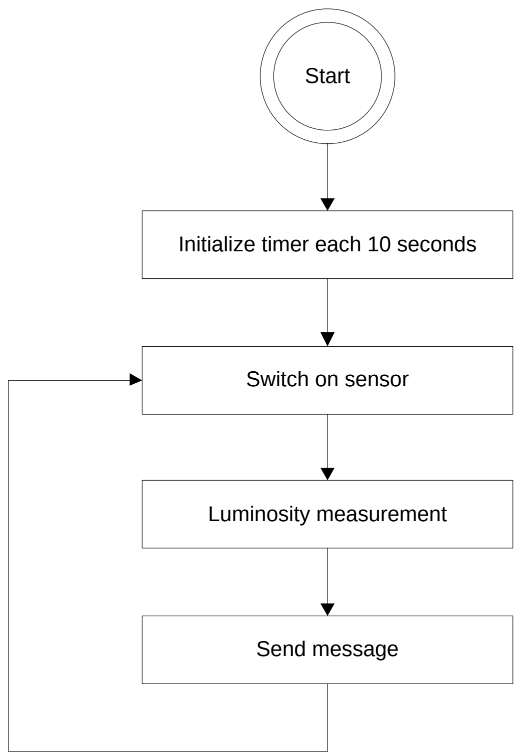


Figure 47. Context collection in a sensor node

The FLoR implementation consists of 4 modules: Unavailability management, Load balancing, Mobility management and Cluster head election (Figure 48).

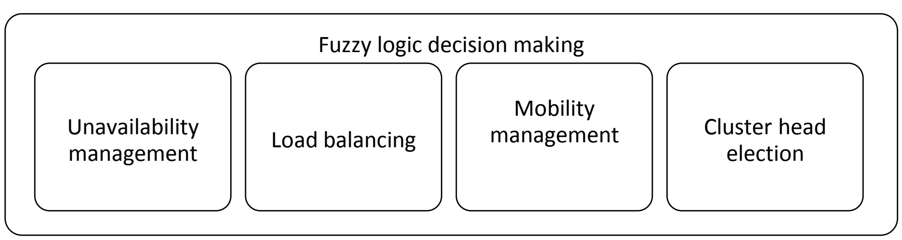


Figure 48. FLoR modules

5.5 Evaluation

In the evaluation of the concept our central question is: does the prototype meet the specifications? How well are individual requirements in the specifications met, and what is the cost for meeting these specifications.

The central functionality is to provide a context-aware system that can be used to alarm others when an abnormal state, like fall, has been detected, and further more to monitor a history of abnormal states to make medical interventions possible. The main requirements are stated as follows:

- detection of position
- detection of the abnormal states like fall
- detection of any abnormal evolution of the health and behavior of the elderly
- privacy preserving

In this objective the context provider and context processor have the key roles. The main question here is: How to provide context information to the application? If the system can deduce the context information, in our system deduced states, the application will be able to analyze this information to lunch an alert or to process their historical evolutions. Due to this importance of the context providing, we evaluated our system to see how it is able to deduce the real state of the context. A high performance context provider shows the high performance of the other low level layers of the system, like: Context provider, System observer, Routing protocol, Load balancing and Failure management etc.

For instance we evaluated the response time of our system, the interaction time between the sensors and the home automation actuators. In this scenario, we use Imote2 motes to sense outdoor luminosity. The prototype consists of 5 fixed Imote2 motes: 1 sensor node that senses the outdoor luminosity, 1 gateway connected to PC and 3 intermediate nodes. The sensed data sent to gateway is processed by middleware, and based on the result, the indoor luminosity will be varied (Figure 49).

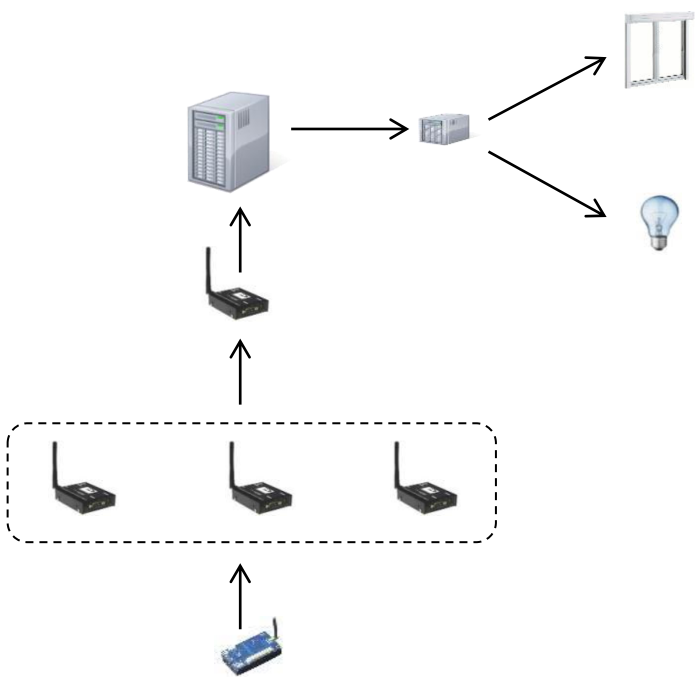


Figure 49. Evaluation scenario

We examined the response time of a context query, which is defined as the interval between the time when the query is issued from an application by receiving the data from the sensor and the time when the application receives the query result as a feedback from the actuator. In this scenario, we use Imote2 motes to sense outdoor luminosity. The sensed data will be sent to the gateway and will be processed by the middleware, and based on the result, the indoor luminosity will be varied.

Table XIV shows the time breakdown of a query in our system. The results presented in this table are the average of 10 measurements.

In the table:

- Data collection is the time spent on getting the context attribute values from the network of Imotes.
- Context processing is the time the context processing of the middleware takes to evaluate the data and generate the query.
- Query generation is the time the system spends on creating the query.
- Reply is the time required to send the query result to the application.

TABLE XIV. RESPONSE TIME

Task	Time (***ms***)	% in total time
Context collection	378	34%
Context processing	112	14%
Query generation	34	4%
Reply	237	48%
Total time	**797**	

In the table we see that, the total response time of a query in our system is less than one second.

The collection of context data from the sensor network takes 34% of the total time and the most time-consuming task among the others is the reply time which takes 48% of total time. The time for context processing takes 14% and the query generation time is just 4% of the total response time. If we use direct command to an actuator we do not have the context collection and processing times. That means by automation the home environment by our system, we add an overhead of 273 ms for context collection and 14% for context processing or 48% of the total response time which seems to be acceptable as a cost of automation of the context processing and control.

Chapter 6 Conclusion

In this chapter we summarize the contributions of our work and discuss possible directions of future work.

6.1 Summary of Contributions

This PhD work addresses the design of telehomecare systems integrating highly dynamic and distributed heterogeneous components such as biomedical and environmental sensors, cameras, etc., which demand a robust QoS method at all layers of the system. Nowadays, many telehomecare system are available, however, it is difficult to find a test-beds that matches exactly the QoS requirements and objectives.

In this thesis, we developed a new concept of telehomecare system for elderly and we proposed a new architecture for this system. In this system we combine home automation sensor and actuator networks with wireless sensor networks which is an emerging technology. Integrating wireless sensor networks can complete the existing home automation sensors for providing precise activity monitoring and significantly reduce the installation cost of the system, because they don't need caballing efforts.

By using this system, MEDeTIC, a non-profit organization, proposes "Vill'Âge®", a new concept of building smart homes for elderly in France.

In a survey done by MEDETIC, a total of 24 elderly dependent people experiencing several types of limitations and living in 3 different residential home care units in France and 13 healthcare professionals were interviewed in order to measure user needs and technology acceptance. Results of this study provided us a great support to improve our system.

The experience gained through the research and development activities during development of this concept, lead us to propose FloR, a Fuzzy Logic based Adaptive Clustering and Routing for Wireless Sensor Networks, and CodaQ, a Context-aware Adaptive QoS middleware.

6.1.1 FLoR - Fuzzy Logic based Adaptive Clustering and Routing for Wireless Sensor Networks

In this thesis, FLoR, a Fuzzy Logic based Adaptive Clustering and Routing for Wireless Sensor Networks, was proposed. FLoR is a multi-objective algorithm. It addresses the major QoS needs in wireless sensor networks: Load balancing, Node mobility management, Communication and node failure. For this reason, 3 new definitions are introduced in this work:

- Mobility: for a node the Mobility is its rate of parent changes.
- Load and load tree: FLoR uses Load tree to balance the load on network.
- Unavailability: a node is Unavailable, when it doesn't answer to a request. The unavailability of a node can be because of its mobility, its failure or failure of the communication link.

In FLoR, each node uses *Energy*, *Mobility*, *Unavailability history* and *Load* of its neighbors and combines them by a fuzzy logic function. The output of this function is Confidence that the node gives to its neighbors.

FLoR is especially effective in networks that use sensor nodes to data collect and in which the data delivery ratio is important and the nodes are mobile, like health monitoring sensor networks. We have shown the efficiency of this algorithm in these networks by simulation compared with LEACH and AODV.

6.1.2 CodaQ - a Context-aware Adaptive QoS middleware

The experience gained through the research and development activities during the 2 first years of the thesis and our propositions for QoS routing algorithm and the healthcare system, showed us the lack of a bridge between the healthcare application and the sensor network.

A bridge is needed to guarantee:

- QoS needs
- Supervision of the system
- Deduce real state of the user and his environment
- Configuration and installation of the sensor and actuators
- Hide the heterogeneity and complexity of the system and sub systems

For these reasons, we developed a context-aware middleware, CodaQ, which has a new architecture. Context Collector, Data management, Context process, System observer, Context abstraction are the main components of the architecture of CodaQ.

In this middleware all data are given a uniform representation, and their level of abstraction is raised through the use of data model definitions. CodaQ introduces fuzzy decision support and comparing to the existing middleware, it addresses a large number of the requirements of telehomecare systems. In CodaQ:

- A fuzzy logic based context-based adaptive QoS method was implemented on the system, in 2 levels: Embedded and State-based QoS.
- To guarantee a continued data providing, creation of virtual sensors is dynamic and based on user location. CodaQ uses virtual sensor to guarantee the spatial coherence of the data.
- The middleware provides necessary contextual information for application layer (Deduced state)

6.1.3 Others

Based on this PhD work, MPIGate, a Multiprotocol Interface and Gateway for healthcare networks, has been developed. MPIGate was awarded in the competition of the Ministry of Higher Education and Research and OSEO 2010.

MPIGate can be used in a telehomecare system to integrate multi-platform systems in a unified framework. Information coming from different systems can be processed and used for having a real state of home environment, human activity, etc. The gateway uses existing and standard industry infrastructure incorporating wireless sensor networks to complete and propose new sensor and mobile applications (Figure 51).

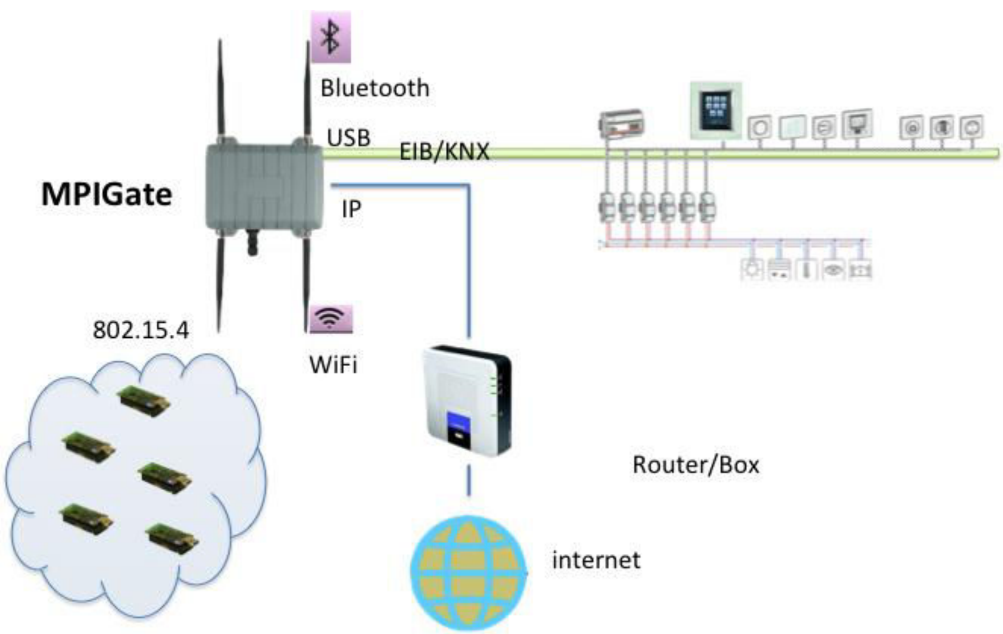


Figure 50. MPIGate in a hetereginous network

MPIGate aims to facilitate the development of applications thanks to the abstraction of sensing data and the facility of access to read and write functions over the devices plugged to the network. The contribution includes modularity and a middleware layer offering modularity and quality of service. The proposed unified syntax of messages facilitates interconnection with other systems, networks and devices.

Figure 51 showe the MPIGate architecture from the functional point of view. Configuration interface: is an interface to configure different parameters of the system. By using this interface we can provide and modify necessary information to process the information in Data processing layer.

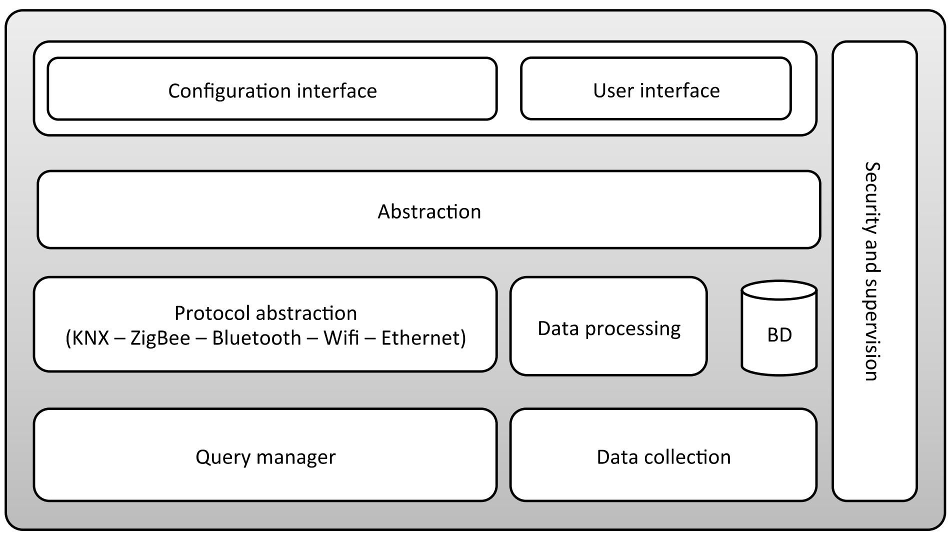


Figure 51. MPIGate components

User interface: provides an Application Programming Interface (API) allowing the programmer of applications to read each sensor of the system. The API allows sending queries to update the sensing values if this is needed. A Graphical Interface is also included to visualize the state of each sensor and actuator, their historical information and the reports generated by the system.

Abstraction: this layer hides the heterogeneity of the system for the users, by providing communication facilities between the interface layer and the other parts of system. The user doesn't need any prior knowledge of the protocols behind the system, greatly facilitating the application developers' task.

Data processing: this layer is the main part of the system. It processes the information collected from the network and save the results in the database.

Query manager: receives the query from user interface, creates the query and sends them to the network. The query might be a command to control an actuator or a request to read a sensor data.

Protocol abstraction: provides necessary information about each communication protocol used in the network.

Security and supervision: it is the responsible of providing the security of the system.

Data collection: collects the data from the network and sends them to the data processing layer. It implements the reading daemons or sniffers for KNX, Bluetooth and IEEE 802.15.4 over the peripherals of MPIGate. The syntax of the messages has been defined in an internal report of INRIA. It is defined as follows:

Node Id, Location, Date, NameOfVariable(i), ValueOfVariable(i), …

Types of variable *i* can be environmental, biometric, equipment state…

6.2 Future work and concluding remarks

We have made contributions to several conception and development phases of distributed telehomecare systems, but there still exist many challenges.

A future work, which has already started, is to test the system in a real world environment by developing an intelligent apartment at Loria (infositue.loria.fr). We will complete the partial implementation of CodaQ in this experimental platform.

The key work is here to design a complete and integrated system able to provide the beneficiary with all communication and medical follow-up and information sharing means necessary to combine wellness, autonomy and self-management, while ensuring psychological and medical follow-up, assistance and social interaction.

This requires a high technological knowledge and a sound collaboration with the users and beneficiaries of such a system which integrates the collection and treatment of data from home and body sensors, in the user's environment and habits, and then produces automated analysis of the medical parameters and emits alerts according to the evolution of the beneficiary's profile. This integration of the solutions that we developed relative to sensors and data correlation systems will be associated to the development of a generic service platform offering access to web services and of a video call center as main communication interface for the beneficiary.

In this respect, one important issue is the development of an activity monitoring application based on multi-sensor data fusion. The CodaQ middleware provides the real state of the context. This information can be used by activity monitoring application to provide a historical and real time analysis.

The performance evaluation of QoS technics is another future work. Issues like the proper and scalable integration, interoperability, the global efficiency and reliability, the scalability and the quality of service are still sources of worry in the concrete technical integration and research phases of a product or service, and source of mistrust among users, patients and many professionals. It shall be studied in details to reach the best balance of innovation, performance and affordability while ensuring a natural integration in the users and beneficiaries' everyday context.

Another future effort focuses on the design of services and interfaces ensuring a maximum level of natural understanding on the way to use them, that is to say with an enhanced effort on the prerequisite and experimentation feedback's of the elderly who are not familiar- and sometimes reluctant with ICT and innovation-based devices and systems.

Finally, the aspects like Security and Privacy were beyond the objectives of this PhD work and didn't considered but can be an interesting open research issue.

References

[1] S. Nourizadeh, C. Deroussent, "Maisons Vill'Âge®: Smart Use of Home Automation Sensor and Actuator Networks for Healthy Aging", ICOST2011, Montreal, Québec

[2] S. Nourizadeh, Y.Q. Song, J.P. Thomesse, C. Deroussent,"A Context-aware Telehomecare System for Senior Citizens", MedeTel 2010, Luxembourg

[3] S. Nourizadeh, C. Deroussent, Y.Q. Song, J.P. Thomesse," Medical and Home automation Sensor Networks for Senior Citizens Telehomecare ", IEEE ICC MAN'2009 Dresden - Germany

[4] S. Nourizadeh, C. Deroussent, Y.Q. Song, J.P. Thomesse, " A Distributed Elderly Healthcare System", MobiHealth2009 , Porto – Portugal

[5] S. Nourizadeh, Y.Q. Song, J.P. Thomesse, X. Seoulchre, "Un Système Distribué de Télémédecine pour Les Seniors" SFTAG09, Troy, France

[6] S. Nourizadeh, Y.Q. Song, J.P. Thomesse, "An Adaptive Hierarchical Routing Protocol for Wireless Ad-hoc Sensor Networks " IEEE NGMAST'09 Cardiff, Wales, UK, 18 Sep. 2009

[7] S. Nourizadeh, Y.Q. Song, J.P. Thomesse, " A Failure Adapted, Load-balanced Distributed Routing for Wireless Ad-hoc Sensor Networks - a Fuzzy Logic based approach " IEEE EUC-2009 Vancouver, CA

[8] S. Nourizadeh, Y.Q. Song, J.P. Thomesse, " A Distributed Fuzzy-based Failure Management for Wireless Sensor Networks " CTRQ2009, Colmar, France

[9] S. Nourizadeh, Y.Q. Song, J.P. Thomesse, " A Location-Unaware Distributed Clustering Algorithm For Mobile Wireless Sensor Networks Using Fuzzy Logic ", FET2007 Toulouse – France

[10] S. Nourizadeh, Y.Q. Song, J.P. Thomesse, "CodaQ – A Context-aware and QoS-aware Middleware for Activity Monitoring", ICOST2011, Montreal, Québec

[11] A. Fleury, N. Noury, "SVM-based multimodal classification of Activities of Daily Living in Health Smart Homes: Sensors, Algorithms and First Experimental Results",IEEE Transactions on Information Technology in Biomedicine 2010.

[12] Gunnar Fagerberg er al. "Platforms for AAL Applications", EuroSSC 2010, LNCS 6446, pp. 177–201, Springer-Verlag Berlin Heidelberg 2010

[13] Xia, F.; Tian, Y.C; Li, Y.J.; Sun, Y.X. "Wireless Sensor/Actuator Network Design for Mobile Control Applications". Sensors 2007, 7(10), 2157-2173.

[14] D. Chen and P.K. Varshney, "QoS Support in Wireless Sensor Networks: A Survey," Proc. Int'l Conf. Wireless Networks (ICWN 04), CSREA Press, 2004, pp. 227–233.

[15] El-Gendy, M. A.; Bose, A.; Shin, K. G. "Evolution of the Internet QoS and support for soft real-time applications". Proceedings of the IEEE 2003, 91(7), 1086-1104.

[16] Chen, D.; Varshney, P. K. "QoS Support in Wireless Sensor Networks: A Survey". In Proc. of the Int. Conf. on Wireless Networks, Las Vegas, USA, June 2004.

[17] Li, Y.J.; Chen, C.S.; Song, Y.-Q.; Wang, Z. "Real-time QoS support in wireless sensor networks: a survey." In Proc of 7th IFAC Int Conf on Fieldbuses & Networks in Industrial & Embedded Systems (FeT'07), Toulouse, France, Nov. 2007.

[18] Bouyssounouse, B.; Sifakis, J. (eds.) "Embedded Systems Design: The ARTIST Roadmap for Research and Development." Lecture Notes in Computer Science 3436, Springer-Verlag, 2005

[19] G. Demiris, "Electronic home healthcare: concepts and challenges", Int. J. Electr. Healthcare 1 (1) (2004) 4–16

[20] M. Chan et al. "A review of smart homes—Present state and future challenges" computer methods and programs in biomedicine - 2 0 0 8

[21] Cheek, P., Nikpour, L., & Nowlin, H.D, "Aging well with smart technology", Nursing Administration Quarterly 2005, 29 (4), 329-338.

[22] Branko G Celler, Nigel H Lovell and Daniel K Y Chan, "The potential impact of home telecare on clinical practice"MJA 1999; 171: 518-521

[23] Marie Chan et al. "Smart homes — Current features and future perspectives" Maturitas October 2009 - volume 64 issue 2 Pages 90-97

[24] MonAmi Project. Mainstreaming on Ambient Intelligence. Funded by the EU 6th framework programme IST (Information Society Technologies). [Online]. Available: http://www.monami.info/

[25] TOPCARE, Fraunhofer Institute for Biomedical Engineering (IBMT), Germany, www.topcare-network.com

[26] http://www.diatelic.com/

[27] Abir Ghorayeb1,Vincent Rialle, Norbert Noury "Modélisation d'un support de communication pour des personnes âgées à domicile", Laboratoire TIMC-IMAG CNRS UMR 5525 Université Joseph Fourier, Grenoble,

[28] Mobihealth project, URL: http://www.mobihealth.org, Accessed in May 2008

[29] V. Shnayder, B.-r. Chen, K. Lorincz, T. Fulfor-Jones, and M. Welsh, "Sensor Networks for Medical Care," Harvard University 2005.

[30] Woo Suk Lee,; Seung Ho Hong "KNX — ZigBee gateway for home automation", Automation Science and Engineering, 2008. CASE 2008

[31] Pradeep Ray; Weerakkody, G. "Quality of service management in health care organizations: a casestudy" Computer-Based Medical Systems, 1999. Proceedings. 12th IEEE Symposium

[32] Boulanger, J. Reerink; Deroussent, C. "Preliminary Based Service Evaluation for Elderly People and Healthcare Professionals in Residential Home Care Units" Digital Society, 2008

[33] A.-K. J.N. and K. A.E. "Routing techniques in wireless sensor networks: a survey". Wireless Communications, IEEE, 11(6):628, 2004.

[34] Z. J. Haas and M. Pearlman, "The zone routing protocol (ZRP) for ad hoc networks," Internet Draft, Tech. Rep.

[35] W. Heinzelman, A. Chandrakasan and H. Balakrishnan,"Energy-efficient communication protocol for wireless microsensor networks," in Proc. of the 33rd Annual Hawaii International Conference on System Sciences (HICSS), Maui, HI, Jan. 2000, pp. 3005 – 3014.

[36] Ossama Younis and Sonia Fahmy "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy- Efficient Approach – HEED", Department of Computer Sciences, Purdue University.

[37] D.J Baker and A. Ephremides, "A Distributed algorithm for Organizing Mobile Radio Telecommunication Networks", in the Proceedings of the 2nd International Conference in Distributed Computer Systems, April 1981.

[38] M. Gerla and J.T.C Tsai, ."Multicluster, mobile, multimedia radio network,. ACM/Baltzer" Journal of Wireless networks, Vol. 1, No. 3, pp. 255-265, 1995.

[39] A.K. Parekh, "Selecting Routers in Ad-Hoc Wireless Networks", Proceedings of the SBT/IEEE InternationalTelecommunications Symposium, August 1994

[40] N. Boughanmi, Y.Q. Song,"A New Routing Metric for Satisfying Both Energy and Delay Constraints in Wireless Sensor Networks"Journal of Signal Processing Systems 51, 137–143, 2008.

[41] L. Yanjun, C. Chung Shue,Y.Q. SONG,W. Zhi,S. Youxian,"Enhancing Real-Time Delivery in Wireless Sensor Networks with Two-Hop Information",IEEE Trans. Industrial Informatics, 2009: 113~122

[42] Indranil Gupta Riordan, D. Srinivas Sampalli, "Cluster-head election using fuzzy logic for wireless sensor networks" IEEE, Proceedings of the 3rd Annual Communication Networks and Services Research Conference (CNSR'05)

[43] Y.M. Tashtoush, M.A. Okour, "Fuzzy Self-Clustering for Wireless Sensor Networks" EUC '08. IEEE/IFIP International Conference Embedded and Ubiquitous Computing, Dec. 2008

[44] H. Chan and A. Perrig, "ACE: An Emergent Algorithm for Highly Uniform Cluster Formation," Proceedings of the First European Workshop on Sensor Networks (EWSN), Berlin, Germany, 2004, pp. 154-171.

[45] Shu-Yin Chiang, Jing-Long Wang,"Routing Analysis Using Fuzzy Logic Systems in Wireless Sensor Networks",Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems, Part II, 2008

[46] L.Khoukhi,S. Cherkaoui,"Experimenting with Fuzzy Logic for QoS Managementin Mobile Ad Hoc Networks",IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.8, August 2008

[47] Xia, F.; Zhao, W.; Sun, Y.; Tian, Y.-C. "Fuzzy Logic Control Based QoS Management in Wireless Sensor/Actuator Networks", Sensors 2007.

[48] Naohiro Hayashibara, Xavier Défago, Rami Yared, Takuya Katayama: "The F Accrual Failure Detector". SRDS 2004: 66-78

[49] M. Weiser, "The Computer of the Twenty-First Century," Scientific American, pp. 94-100, September 1991.

[50] D. Tennenhouse, "Proactive Computing," Communications of the ACM, vol. 43, no. 5, pp. 43-50, 2000.

[51] D. Estrin, R. Govindan, and J. Heidemann. "Embedding the Internet: Introduction," Communications of the ACM, vol. 43, no. 5, pp. 38-42, 2000.

[52] K. Marzullo, "Tolerating failures of continuousvalued sensors," ACM Transactions on Computer Systems, vol. 8, no.4, pp. 284-304, 1990.

[53] D.N. Jayasimha, "Fault tolerance in multi-sensor networks," IEEE Transactions on Reliability, vol. 45, no.2, pp. 308-15, June 1996.

[54] Perkins, et. al. "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, July 2003

[55] D. C. Daly and A. P. Chandrakasan, "An energy-efficient OOK transceiver for wireless sensor networks," *IEEE J. Solid-State Circuits*, vol. 42, no. 5, pp. 1003-1011, May 2007.

[56] K. JeongGil, L. Chenyang, M. Srivastava, J. Stankovic, A. Terzis, M. Welsh, "Wireless Sensor Networks for Healthcare", Proceedings of the IEEE, Vol.98, p. 1947 – 1960, (2010).

[57] Dey, Anind K. (2001). "Understanding and Using Context". Personal Ubiquitous Computing 5 (1): 4–7. doi:10.1007/s007790170019

[58] Anind K. Dey and Gregory D. Abowd, "Towards a Better Understanding of Context and Context-Awareness", CHI 2000 Workshop on the What, Who, Where, When, Why and How of Context-Awareness, April 1-6, 2000

[59] T. Buchholz, A. Küpper, and M. Schiffers, "Quality of context: What it is and why we need it", The Workshop of the HP OpenView University Association 2003HPOVUA 2003), Geneva, 2003.

[60] K. Sheikh, M. Wegdam, M. van Sinderen, "Middleware Support for Quality of Context in Pervasive Context-Aware Systems", PerCom Workshops '07, Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2007.

[61] Matthias Baldauf, Schahram Dustdar and Florian Rosenberg, "A survey on context-aware systems", Int. J. Ad Hoc and Ubiquitous Computing, Vol. 2, No. 4, 2007.

[62] Abdulrazak, B., Chikhaoui, B., Gouin-Vallerand, C., Fraikin, B.: "A Standard Ontology for Smart Spaces." International Journal of Web and Grid Services Vol. 6 (3) 2010: pp. 244-268.

[63] Markus C. Huebscher, Julie A. McCann, "An adaptive middleware framework for context-aware applications", Personal and Ubiquitous Computing 10(1): 12-20 (2006)

[64] L. Coyleet al. " Sensor Fusion-Based Middleware for Assisted Living" IOS Press, 200.

[65] Henricksen, K., Indulska, J., McFadden, T., & Balasubramaniam, S. (2005). "Middleware for Distributed Context-Aware Systems". OTM Confederated International Conferences (pp. 846-863). Springer-Verlag.

[66] De Almeida, D. R., De Souza Baptista, C., Da Silva, E. R., Campelo, C. E., De Figueiredo, H. F., & Lacerda, Y. A. (2006). « A Context-Aware System Based on Service-Oriented Architecture." Proceedings of the 20th international Conference on Advanced information Networking and Applications - Volume 1 (Aina'06) - Volume 01 (pp. 205-210). IEEE Computer Society.

[67] Gua, T., Punga, H. K., & Zhang, D. Q. (2008). "A service-oriented middleware for building context-aware services". Journal of Network and Computer Applications , 28 (1), 1-18.

[68] H.-P. Dommel, R. Wagner, A. Doran, and R. Edwards, "A Middleware Framework for the Adaptive Home", Third International Conference on Smart homes and Health Telematics (ICOST'2005), Sherbrooke, Québec, Canada, July 4-6, 2005

[69] W3C. Resource Description Framework (RDF). http://www.w3.org/RDF/

[70] B. McBride. "Jena: Implementing the RDF model and syntax specification". In Proceedings of the 2nd International Workshop on the Semantic Web, Hong Kong, may 2001

[71] D. Garlan, at al, "Project Aura: Towards Distraction-Free Pervasive Computing," in IEEE Pervasive Computing, vol. 1, 2002, pp. 22-31.

[72] Bellavista, P., Corradi, A., Montanari, R., & Stefanelli, C. (2003). « Context-Aware Middleware for Resource Management in the Wireless Internet". IEEE Trans. Software Eng. , 29 (12), 1086-1099.

[73] Capra, L. and Emmerich, W. and Mascolo, C. (2003) "CARISMA: Context-Aware Reflective mIddleware System for Mobile Applications". IEEE Transactions on Software Engineering, 29 (10). pp. 929-945.

[74] Barton, John; Kindberg, Tim, "The Cooltown user experience", CHI2001 Workshop on Building the Ubiquitous Computing User ExperienceHPL-2001-22

[75] Philippe Debaty, Patrick Goddi, Alex Vorbau, "Integrating the Physical World with the Web to Enable Context-Enhanced Mobile Services", MONET 2005 Vol. 10, 385-394

[76] C. Sørensen et al., "A context-aware middleware for applications in mobile Ad Hoc environments" 2nd workshop on Middleware for pervasive and ad-hoc computing (MPAC '04), ACM, New York, NY, USA, 107-110

[77] M. Roman, C. Hess, R. Cerqueira, A. Ranganathan, R.H. Campbell, K. Nahrstedt, "Gaia: A Middleware Infrastructure for Active Spaces", IEEE Pervasive Computing, Vol. 1, No. 4, October-December, 2002

[78] Roman, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R., & Nahrstedt, K. (2002, Oct-Dec). "Gaia: A middleware infrastructure to enable active spaces". IEEE Pervasive Computing , pp. 74--83.

[79] A. Ranganathan, J. Al-Muhtadi, S. Chetan, R. Campbell, and M. D. Mickunas, "MiddleWhere: A Middleware for Location Awareness in Ubiquitous Computing Applications," 5th International Middleware Conference (Middleware 2004), 2004.

[80] A. Chan and S.-N. Chuang, "MobiPADS: A Reflective Middleware for Context-Aware Mobile Computing," IEEE Trans. Software Eng., vol. 29, no. 12, pp. 1072-1085, Dec. 2003.

[81] Klara Nahrstedt, Dongyan Xu, Duangdao Wichadakul, Baochun Li, "QoS-Aware Middleware for Ubiquitous and Heterogeneous Environments", IEEE Communications Magazine, 2001, vol. 39, pages 140-148

[82] Grégory Nain, Olivier Barais, Régis Fleurquin, Jean-Marc Jézéquel, " EntiMid : un middleware aux services de la maison", In 3ème Conférence Francophone sur les Architectures Logicielles (CAL'O9), Nancy, France, Mars 2009

[83] Thomas Vergnaud, Jerome Hugues, Laurent Pautet, and Fabrice Kordon. Polyorb: "A schizophrenic middleware to build versatile reliable distributed applications". LNCS, 3063:106–119, 2004.

[84] The UPnP Forum. http://www.upnp.org.

[85] Fran¸cois Jammes, Antoine Mensch, and Harm Smit. "Service-oriented device communications using the devices profile for web services". In MPAC '05: Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing, pages 1–8, New York, NY, USA, 2005. ACM.

[86] Ning Ye, Ruchuan Wang, Shou-ming Ma, Zhili Wang: "Fuzzy Logic Based Middleware Approach for Context Processing". JDCTA 3(3): 36-41 (2009)

[87] Padovitz, A., Loke, S.W., Zaslavsky, A., Burg, B. and Bartolini, C. "An Approach to Data Fusion for Context Awareness". Fifth International Conference on Modelling and Using Context, CONTEXT'05, Paris, France, July 2005, CONTEXT 2005.

[88] Hans W. Guesgen and Stephen Marsland. “Spatio-Temporal Reasoning and Context Awareness”, pages 609-634. Springer, Berlin, Germany, 2010.

[89] Richard Beckwith, "Designing for Ubiquity: The Perception of Privacy", IEEE Pervasive Computing, v.2 n.2, p.40-46, April 2003

[90] T. Heiber and P. J. Marron, “Exploring the relationship between context and privacy,” Privacy, Security and Trust within the Context of Pervasive Computing,Kluwer International Series in Engineering and Computer Science, 2005.

[91] J. I. Hong and J. A. Landay, “An architecture for privacy-sensitive ubiquitous computing,” in Proc. 2nd International Conf. on Mobile systems, applications,and services (MobiSys ’04), 2004.

[92] H. Alemdar, C. Ersoy,"Wireless sensor networks for healthcare: A survey",Computer Networks,Volume 54, Issue 15, 28 October 2010, Pages 2688-2710

[93] J. Ko et al., "Wireless sensor networks for healthcare",IEEE proccedings, Vov. 2010,Volume 98(11), 1947-60

[94] L. Zadeh, “Fuzzy sets, Information and Control”, 1965, 8(3):338-353

[95] L. Zadeh, “Communications of the ACM”, 1984, Volume 27 Number 4

[96] L.A. Zadeh, “Fuzzy sets as a basis for a theory of possibility,” Fuzzy Sets and Systems 1, pp. 3-28, 1978.

[97] L.A. Zadeh,”From imprecise to granular probabilities”, Fuzzy Sets and Systems, 370–374, 2005.

[98] L.A. Zadeh, “Toward a perception-based theory ofprobabilistic reasoning with imprecise probabilities”, J. Statist. Plann. Inference 105 -2002.

[99] Wang Lie-Xin, “A course in fuzzy systems and control”, Prentice Hall, 1996.

[100] Hao Ying, "Fuzzy Control and Modeling: Analytical Foundations and Applications", Chapter1, Wiley-IEEE Press, August 2000

[101] L.A. Zadeh, “Concept of linguistic variable Part1”, 1975

[102] L.A. Zadeh, “Outline of a New Approach to the Analysis of Complex System and Decision Processes,” *IEEE Trans. on Systems, Man, and Cybernetic*s, SMC-3, pp.28-44, 1973.

[103] L.A. Zadeh, "From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions" IEEE AIP conference proceeding, June 2001 -- Volume 573, pp. 36-58

[104] L.A. Zadeh, "A fuzzy-set-theoretic interpretation of linguistic hedges," *J. of Cybernetics* 2, pp. 4-34, 1972.

[105] L.A. Zadeh, "Fuzzy Logic = Computing with Words," *IEEE Transactions on Fuzzy System*s, Vol. 4, pp. 103-111, 1996.

[106] D.Dubois, H.Prade,"What are fuzzy rules and how to use them", Fuzzy Sets and Systems, Volume 84, Number 2, pp. 169-185(17), September 1996

[107] L.A. Zadeh, "The calculus of fuzzy if/then rules", AI Expert, 7(3), 23-27, 1992.

[108] L.A. Zadeh, "Soft computing and fuzzy logic", IEEE Software, November, 48-56, 1994.

[109] E.H. Mamdami, S. Assilina, "An experiment in linguistic synthesis with a fuzzy logic controller", International Journal of Man-Machine Studies, 1975;

[110] M. Sugeno, "Industrial applications of fuzzy control", Elsevier Science Pub. Co., 1985;

Appendix A Fuzzy Logic

Introduction

Fuzzy logic is an extension of Boolean logic dealing with the concept of partial truth which denotes the extent to which a proposition is true. As we know classical logic holds that everything can be expressed in binary terms (0 or 1, black or white, yes or no), fuzzy logic replaces Boolean truth values with a degree of truth. Degree of truth is often employed to capture the imprecise modes of reasoning that play an essential role in the human ability to make decisions in an environment of uncertainty and imprecision. As a human being, we have an incredible ability to communicate skills simply and accurately by using vague linguistic rules. For example, in sensor network domain, we can propose the rules like:

- *If the A is **far** from B and the Energy of A is **Low**, then A is **not Good** cluster-head for B*
- *If the A is **close** B and the Energy of A is **High**, then A is a **Good** cluster-head for B*

to organize the network into the clusters. These are linguistic rules and are very descriptive and make perfect sense to a human but are difficult to translate into a language a computer can understand. Words like "far," "very close,", "Low" and “Good” do not have sharp, well-defined boundaries, and when we attempt to describe them in code the result often looks clumsy and artificial. We may encode the term "Distance" as:

- Close = A is between 0 meters and 10 meters from B.
- Medium = A is between 10 meters and 20 meters from B.
- Far = A is greater than 20 meters from B.

But what if A is 19.99 meters away from B? Using these descriptives to represent distances, a computer will put A firmly in the " Medium distance " slot, even though the addition of a couple more centimeters will transform it to being far away! Of course, it's possible to reduce the effect of this problem by creating smaller and smaller intervals, but the underlying problem remains because the distance terms are still represented by discrete intervals.

Compare this to how a human reasons. When considering linguistic terms such as "far" and "close" a human being is able to place vague boundaries on those terms and allow a value to be associated with a term to a *matter of degree*. When A is 19.99 meters away from B, a human will regard it to be *partly* associated with the term "Medium distance" but *mostly* with the term "far distance." In this way humans perceive the distance quality of A *gradually* shifting between linguistic terms instead of changing abruptly, allowing us to reason accurately about linguistic rules such as the ones given for choosing a Cluster-head.

The concept of fuzzy logic was proposed by professor Zadeh, professor at the University of California at Berkley. In 1964, professor Zadeh started wondering, if there wasn't a better logic to use in machinery. He had the idea that if you could tell an air-conditioner to work a little faster when it gets hotter, or similar problems, it would be much more efficient than having to give a rule for each temperature. Fuzzy logic was started by introducing the theory of fuzzy sets in the 1965 paper "Fuzzy Sets" [94]. He presented fuzzy logic as a multivalued logic and a way of processing data by allowing partial set membership.

> *"In classical two-valued systems, all classes are assumed to have sharply defined boundaries. So either an object is a member of a class or it is not a member of a class. Now, this is okay if you are talking about something like mortal or not mortal, dead or alive, male or female, and so forth. These are examples of classes that have sharp boundaries. But most classes in the real world do not have sharp boundaries. For example, if you consider characteristics or properties like tall, intelligent, tired, sick, and so forth, all of these characteristics lack sharp boundaries."*
>
> *[95]L. Zadeh- Communications of the ACM, April 1984, Volume 27 Number 4*

Fuzzy logic enables a computer to reason about linguistic terms and rules in a way similar to humans. Concepts like "far" or "slightly" are not represented by discrete intervals, but by fuzzy sets, enabling values to be assigned to sets to a matter of a degree — a process called fuzzification. Using fuzzified values computers are able to interpret linguistic rules and produce an output that may remain fuzzy or can be defuzzified to provide a crisp value. This is known as fuzzy rule-based inference, and is one of the most popular uses of fuzzy logic. See Figure 50.

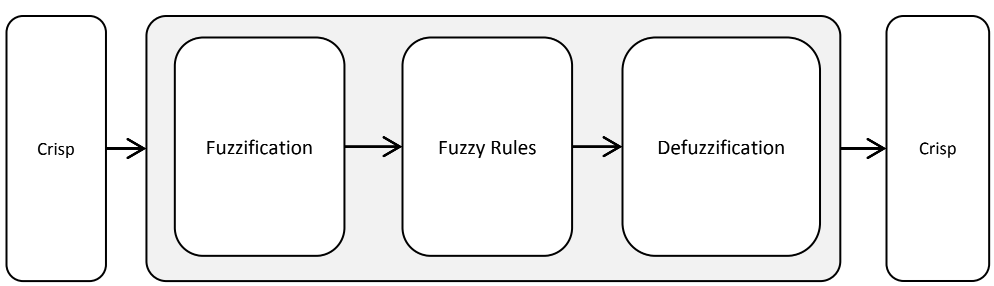


Figure 52. Fuzzy rule-based inference

Note:

When talking about Fuzzy logic, many persons think that it is synonym of probability, but Fuzzy logic and probability refer to different kinds of uncertainty. Fuzzy logic is specifically designed to deal with imprecision of facts (fuzzy logic statements), while probability deals with chances of that happening (but still considering the result to be precise).

Lotfi Zadeh, the creator of fuzzy logic, argues [96-97] that fuzzy logic is different in character from probability, and is not a replacement for it and also probability is a sub-theory of fuzzy logic, as probability only handles one kind of uncertainty. He has created a fuzzy alternative to probability, which he calls possibility theory. The probability of an event is generally represented as a real number between 0 and 1, inclusive. An impossible event has a probability of exactly 0, and a certain event has a probability of 1, but the converses are not always true: probability 0 events are not always impossible, nor probability 1 events certain. The rather subtle distinction between "certain" and "probability 1" is treated at greater length in the article on "almost surely".

What is Fuzzy Logic?

In this section the fuzzy logic will be detailed. The concept of fuzzy sets, fuzzy membership, fuzzification, fuzzy rules and defuzzification process will be discussed in the section. But before starting a detailed description of fuzzy logic, we will take a look to the mathematics of crisp sets. This will help us to understand fuzzy sets, which are the base of fuzzy logic.

Crisp sets

Crisp sets are the mathematical sets which have clearly defined boundaries that means an *object* or an *element* either completely belongs to a set or it doesn't. This is fine for many problems since many objects can be precisely classified. The domain of all elements a set belongs to is called the *universe of discourse.* Any crisp set A, subset of universe of discourse X, is characterized by a membership function $\mu: X \rightarrow \{0, 1\}$. For an element x, its membership value, $\mu A(x)$ is 1 if $x \in A$ and is 0 if $x \notin A$ which means each statement about a classical set has two truth values: True and False (or 0 and 1).

The white rectangle of Figure 51 represents the universe of discourse of xbow's sensor boards. The orange circle presents the set of sensors with temperature sensing capabilities, the green circle, is the set of accelerometers and the blue one denotes the motion detectors. By using mathematical notation, we can present these sets, as:

- Temperature = {MTS300, MTS310, MTS400, ITS400}
- Accelerometer = { MTS310, MTS400, ITS400}
- Motion = {IMB400}

As is evident, the degree of membership of a number to a crisp set is either true or false, 1 or 0. The IMB is 100 percent a Motion sensor and 0 percent an Accelerometer. In classical set theory all the integers are black and white in this way — they are members of one set to a degree of 1 and to the other to a degree of 0. It's also worth highlighting that an element can be contained in more than one crisp set. For example, the MTS310 is a member of the set of Accelerometer sensors and the set of temperature sensors and in all these sets its degree of membership is 1.

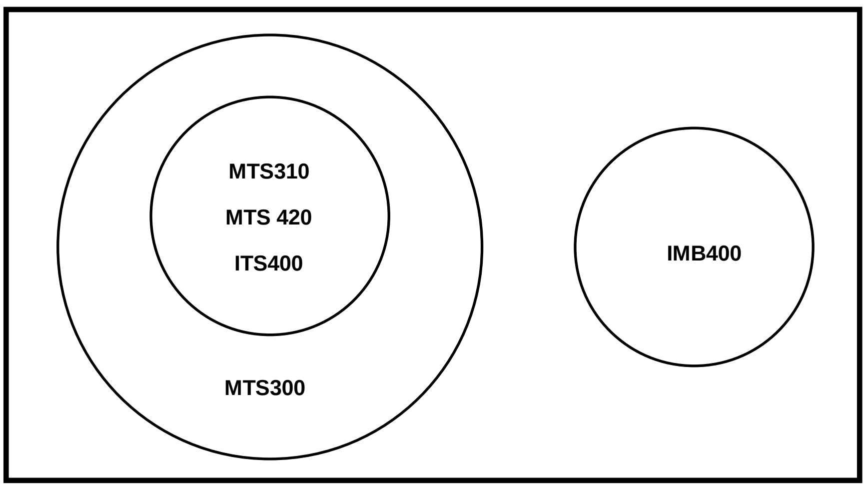


Figure 53. Crisp set

Between the number of operations that can be performed on sets, union, intersection, and complement are the most common.

- The union of two sets is the set that contains all the elements from both sets. The union operator is usually written using the symbol ∪ which is equivalent to ORing the sets together. In our example, the union of the two sets Temperature and Motion can be written as: T∪M = {MTS300, MTS310, MTS400, ITS400, IMB400}

- The intersection of two sets, written using the symbol ∩, is the set containing all the elements present in both sets and is equivalent to ANDing the sets together. For the two sets Temperature and Accelerometer, their intersection is written as: T∩M = { MTS310, MTS400, ITS400}
- The complement of a set is the set containing all the elements in the universe of discourse not present in the set. In other words, it is equivalent to NOT. For example if the universe of discourse is: UoD = {MTS300, MTS310, MTS400, ITS400, IMB400}, then:
- *Temperature*' = {IMB400}
- *Accelerometer*'= {IMB400, ITS400}
- *Motion'* = {MTS300, MTS310, MTS400, ITS400}

Fuzzy sets and membership functions

> *"A fuzzy set is a class of objects with a continuum of grades of membership. Such a set is characterized by a membership (characteristic) function which assigns to each object a grade of membership ranging between zero and one."*
>
> *[95]L. Zadeh- Fuzzy sets, Information and Control, 1965, 8(3):338-353*

Fuzzy logic is started by introducing the concept of fuzzy set. A fuzzy set is a set without a crisp, which means without clearly defined boundary. It can contain elements with only a partial degree of membership. Fuzzy sets have membership properties defined between 0 and 1.

In fuzzy logic systems, a fussy set A, sub set of universe of discourse X, is presented by a membership function μ: X → [0, 1] and for an element *x*, μ(x) is its degree of membership in A; which means, there are more than two truth values for each statement. There may be a finite or even an infinite number of truth values.

Let's continue with some examples. In a wireless sensor network, for distance of the sensors from base station, the concept "near" can be an example. Figure 52 shows an example of a crisp set for "near". By using this crisp set, all the sensors with a distance from 0 to 15 m from base station have a membership with value of 1, and for the other sensors, the membership value is 0. But how can we explain a value 1 for a sensor with distance of 14.9 and a value of 0 for a sensor with distance of 15.1?

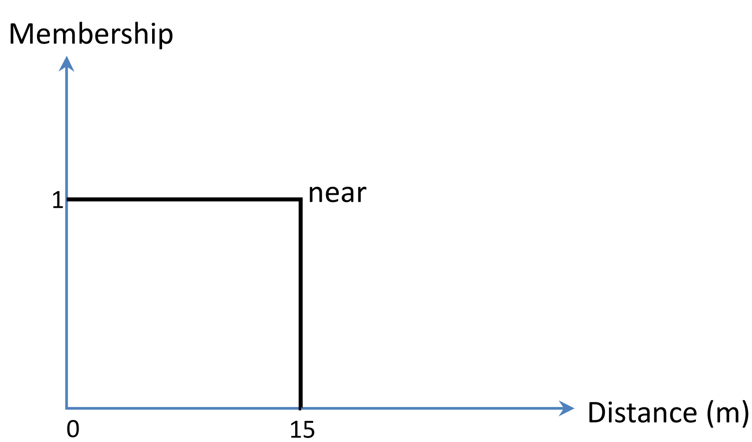


Figure 54. a crisp set for "near" distance

The energy (battery charge) of sensor is another example. In a sensor network, for a specific sensor, its energy level is precise and let's set A be the set of *medium energy level sensors*. But relating a particular energy level to the set A, involves confusing and difficulties. The question is "what energy level is medium?" assume that the initial value of battery charge of a sensor is 3 volts.

To answer this question we can say all the sensors with 1.5 volts of battery charge belong to the A (see figure 53). But what we can say about a sensor with 1.4 or 1.6 volts of charges?

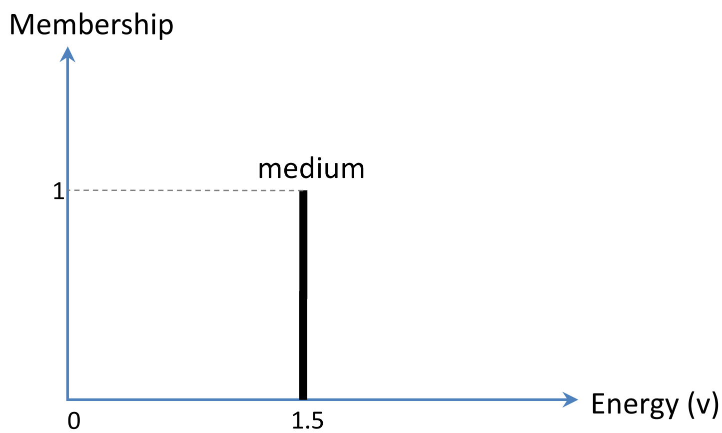


Figure 55. Figure 1.a crisp set for "medium" battery charges

By using fuzzy sets, we can answer these questions. A fuzzy set is defined by a *membership function*.The membership function of a fuzzy set corresponds to the indicator function of the classical sets. It is a curve that defines how each point in the input space is mapped to a membership value or a degree of truth between 0 and 1. The input space is sometimes referred to as the universe of discourse [100].

These functions can be any arbitrary shape but are typically triangular or trapezoidal. Figure 54 shows a few examples of membership functions. They define a gradual transition from regions completely outside the set to regions completely within the set, thereby enabling a value to have *partial membership* to a set. This is the base of fuzzy logic.

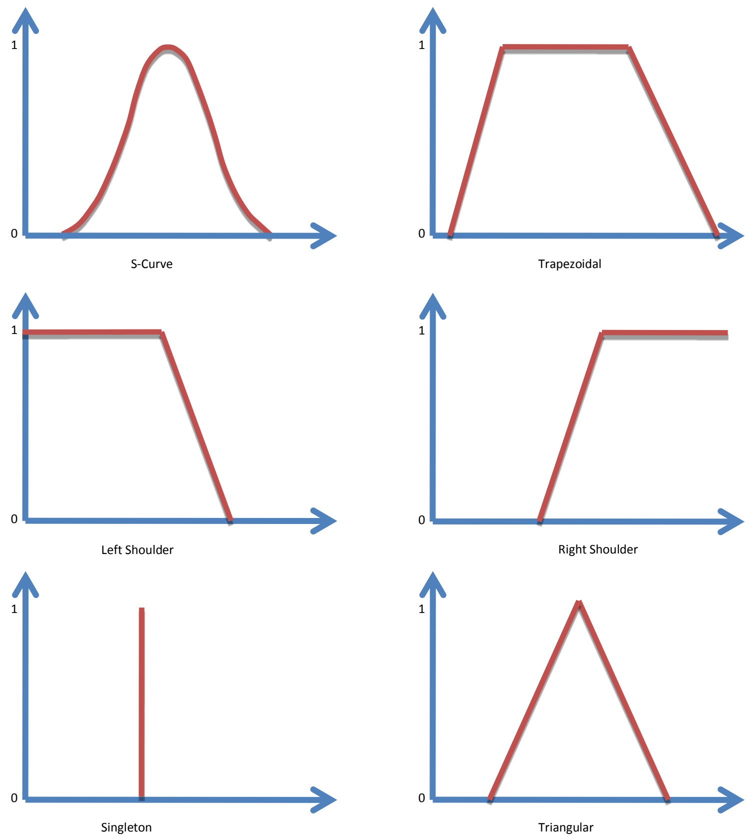


Figure 56. Fuzzy membership functions

Figure 55 shows a possible fuzzy set to describe concept of "near". As we can see in this figure, all the sensors of our network belong to the set "near" by a membership degree between 0 and 1. A sensor with distance of 1(meters from base station has a membership degree of 0.70 and for another one by a distance of 20 meters, the membership degree is 0.20. all the sensors with distance≤10, have 1 membership degree.

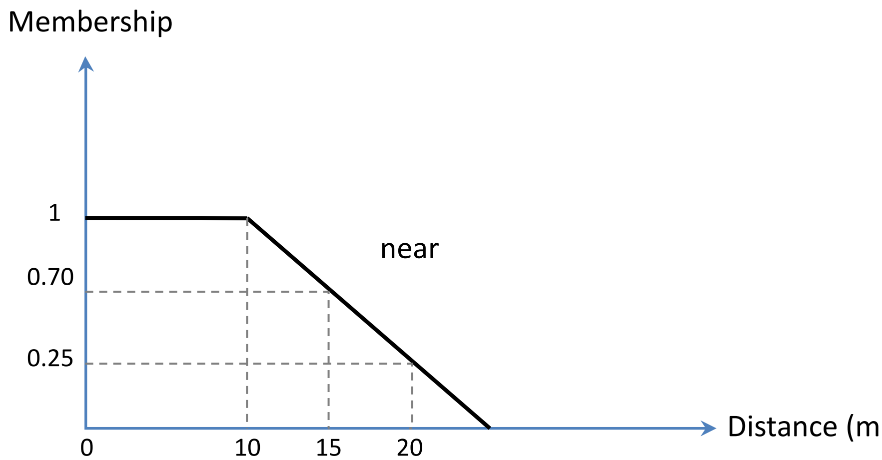


Figure 57. fuzzy set for "near" distance

We can say that a fuzzy set consist of a universe of discourse and a membership function that maps every element in the universe of discourse to a membership value between 0 and 1 [101]. For our second example, the battery charge of the sensors, on possible fuzzy set can be the fuzzy set presented by Figure 56. In this case all the sensors with battery charge in [1, 2] are belong to the medium set by a membership degree of [0, 1] and for the others, the membership degree is 0.

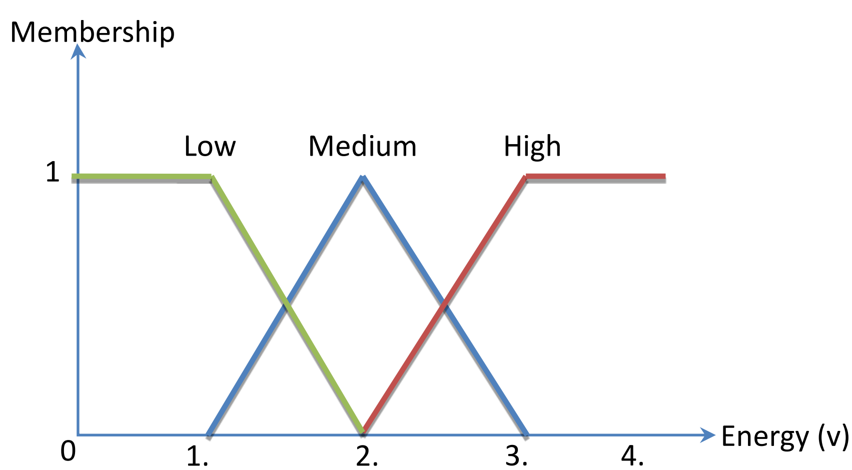


Figure 58. fuzzy membership function for battery energy

Fuzzy sets operators

The Intersections, unions, and complements of fuzzy sets are the same as for crisp sets. The fuzzy intersection operation is equivalent to the AND operator. The result of intersection of two or more fuzzy sets is another fuzzy set. For two fuzzy sets A and B, with the membership functions: μA (x) and μB (x):

$$\mu A \cap B\ (x) = \min\{\mu A\ (x)\ ,\ \mu B\ (x)\}\ ,\ x \in X$$

The fuzzy intersection operator can be represented as the algebraic product of two fuzzy sets A and B, which is defined as the multiplication of their membership functions: $\mu A \cap B\ (x) = \mu A\ (x) \mu B\ (x),\ x \in X$

Figure 57 illustrates the fuzzy set of sensor nodes whose Energy level is medium and high.

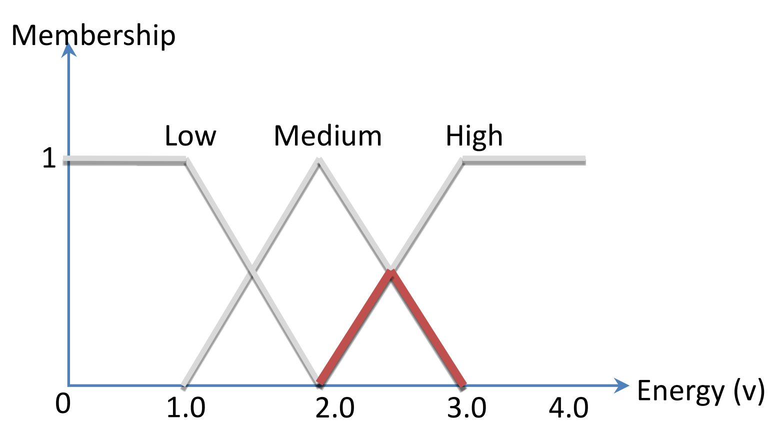


Figure 59. and" operator

The fuzzy *union* operation is equivalent to the OR operator. For two fuzzy sets A and B, with the membership functions: $\mu A\ (x)$ and $\mu B\ (x)$:

$$\mu A \cup B\ (x) = \min\{\mu A\ (x)\ ,\ \mu B\ (x)\}\ ,\ x \in X$$

As we said for intersection, here also, the result of union of two or more fuzzy sets is another fuzzy set. It can also be represented as the algebraic sum of two fuzzy sets A and B, which is defined as:

$$\mu A \cap B\ (x) = \mu A\ (x) + \mu B\ (x) - \mu A\ (x) \mu B\ (x)\ ,\ x \in X$$

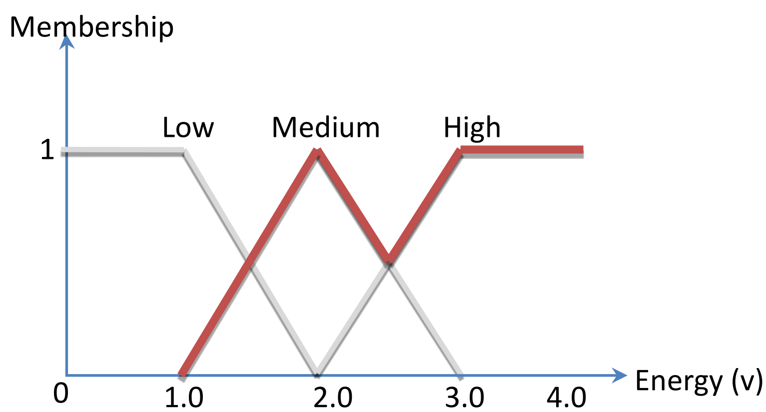


Figure 60. “or” operator

Figure 58 illustrates the fuzzy set of sensor nodes whose Energy level is medium or high. The *fuzzy complement* (fuzzy NOT operation) applied to the fuzzy set A with the membership function $\mu A\ (x)$ is: $\mu \hat{A}\ (x) = 1 - \mu A\ (x), x \in X$

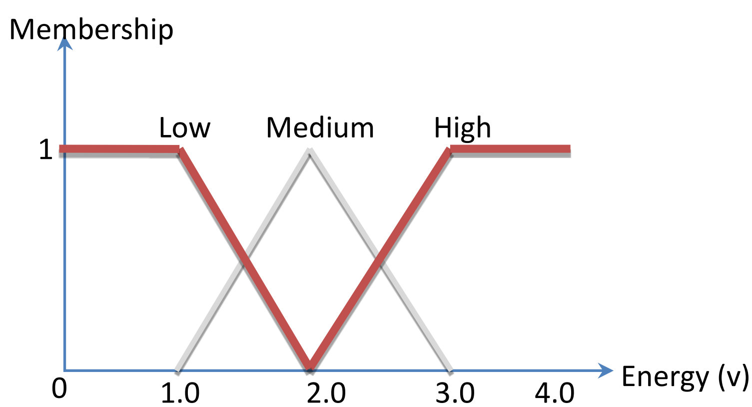


Figure 61. “not” operator

Figure 59 illustrates the fuzzy set of sensor nodes whose *Energy* level is medium or high.

Linguistic variables

> *"A linguistic variable is a variable whose values are not numbers but words or sentences in a natural or artificial language."*
>
> *[102]Zadeh, 1975, Concept of linguistic variable Part1*

In a paper published in 1973, [103] Prof. Zadeh, introduced the concept of a linguistic variable, that is, a variable whose values are words rather than numbers. The concept of a linguistic variable has played and is continuing to play a pivotal role in the development of fuzzy logic and its applications [107].

A fuzzy linguistic variable is the composition of one or more fuzzy sets to represent a concept. Given our earlier example, the sets High, Medium, and Low are members of the fuzzy linguistic variable Energy. This can be written as: Energy = {*High*, *Medium*, *Low*}; Here the label "*High*" is a linguistic value (Fuzzy value) for linguistic variable "Energy".

More formally, a linguistic variable is characterized by a quintuple, say [χ , T(χ), U, G, M], in which χ stands for the name of the variable, T(χ) denotes the term set of χ , with each value being a fuzzy variable denoted generically by X, and ranging over a universe of discourse U that is associated with the variable u. G is a syntactic rule (usually a grammar) for generating names, X, of values of χ , and M is a semantic rule for associating with each X its meaning, M(X), which is a fuzzy subset of U [102].

Hedges

An important ability of fuzzy systems is to modify the fuzzy rules by defining "hedges" or "modifiers". Hedges are operations to reduce the distance to natural language [105, 106]. These operators change the meaning of a term or a fuzzy set, by modifying its membership function.

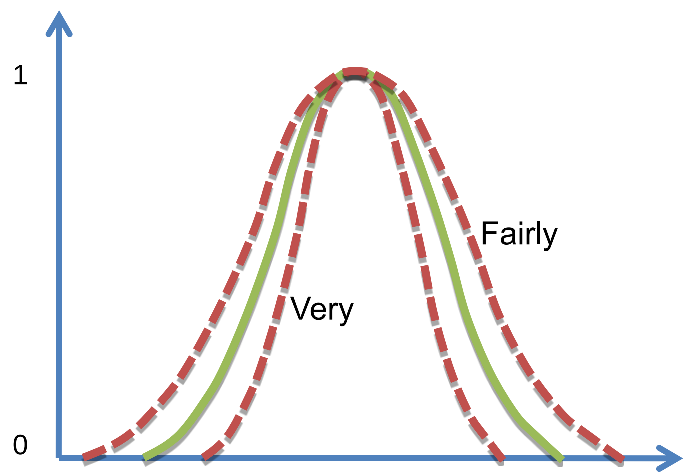


Figure 62. "Very" and "Fairly" hedges

They also allow the generation of fuzzy statements through mathematical calculations. Two commonly used hedges are VERY and FAIRLY. For a fuzzy set A, VERY modifies it like so: $F_{Very}(A) = (F_A(x))^2$

In other words, it results in the square of the membership degree. FAIRLY modifies a fuzzy set by taking the square root of the membership degree, like so: $F_{Fairly}(A) = \sqrt{F_A(x)}$

Figure 60 illustrates Very and Fairly hedges. As an example, by VERY hedge, we can transform the statement "This energy of the sensor 'A' in Low" to "This energy of the sensor 'A' in VERY Low". If F presents the Energy membership function, and if $F_{Low}(A)= 0.2$, then $F_{VeryLow}(A)= 0.4$.

If-Then rules

If/then rules offer a convenient format for expressing pieces of knowledge. But it is just a format which can cover different intended semantics and uses. The consequent of a rule may be qualified with various modalities expressing for instance, strong ones such as certainty, obligation, or weaker ones such as possibility or feasibility.

Rules may have implicit exceptions, or may universally hold. Depending on their interpretation, rules have to be represented and processed in a specific way at the inference level. Strangely enough, in spite of the acknowledged importance of fuzzy rules and of the great number of works in fuzzy set based approximate reasoning, there has been very little concern until now about the possible intended meanings of a fuzzy rule, although fuzzy rules seem to play an important role in our thinking mechanisms.[107]

In a IF-THEN rule the IF part is called the "antecedent" and the THEN part is called the "consequent". Fuzzy if/then rules are rules whose antecedents, consequences or both are fuzzy rather than crisp. The most common and widely used interpretation considers a fuzzy rule "if x is A then y is B" as a fuzzy point $A \times B$ and a collection of fuzzy rules "if x is Ai then y is Bi" i =1, …, n as a fuzzy graph providing a rough description of the relation between x and y [108, 109].

How does it work?

Generally, a fuzzy system consists of three main stages: input, processing, and output stage. The input stage maps the crisp inputs to the fuzzy values. The processing stage invokes each appropriate rule and generates a corresponding result. It then combines the results. Finally, the output stage converts the combined result back into a specific output value. The processing stage which is called inference engine is based on a collection of IF-THEN rules. There are two common inference processes.

First is called Mamdani's fuzzy inference method proposed by Ebrahim Mamdani [109] and the other is Sugeno method [110]. These two methods are the same in many respects, such as the procedure of fuzzifying the inputs and fuzzy operators.

The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant but Mamdani's inference expects the output membership functions to be fuzzy sets. Due to its simple structure of 'min-max' operations, Mamdani's method is the most commonly used.

Fuzzy Inference Systems

Generally a fuzzy inference system consists of four modules (Figure 61):

- Fuzzification module: transforms the system inputs, which are crisp numbers, into fuzzy sets. This is done by applying a fuzzification function.
- Knowledge base: stores IF-THEN rules provided by experts.

- Inference engine: simulates the human reasoning process by making fuzzy inference on the inputs and IF-THEN rules by consulting an existing knowledgebase.
- Defuzzification module: transforms the fuzzy set obtained by the inference engine into a single crisp value.

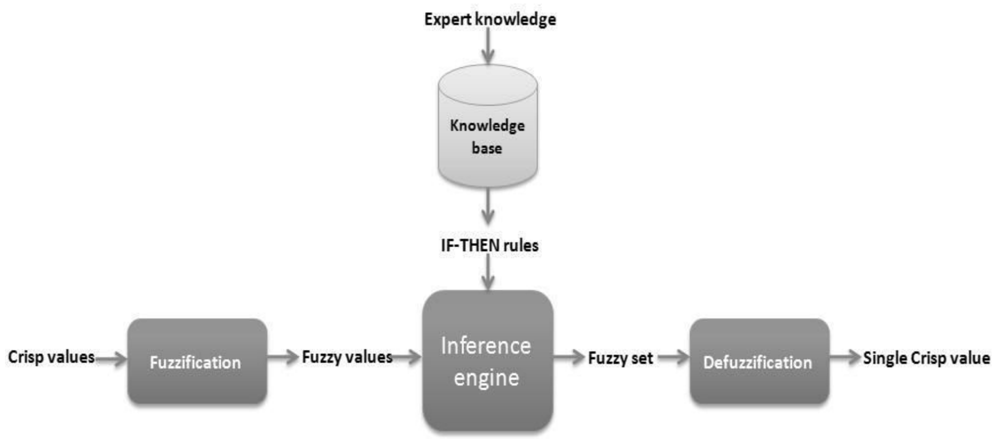


Figure 63. Fuzzy Inference System

An inference engine tries to process the given inputs and produce an output by consulting an existing knowledgebase. The three steps toward a fuzzy inference engine are, Applying Fuzzy Operators, Applying Implication Methods, and Aggregating All Outputs.

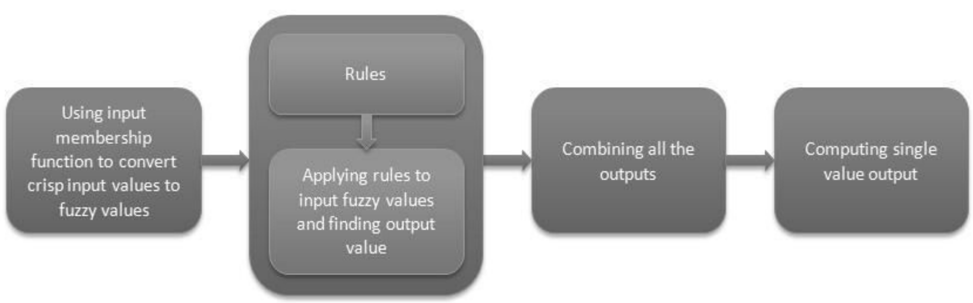


Figure 64. Steps is a typical fuzzy Inference engine

Once the inputs have been fuzzified, the degree to which each part of the antecedent has been satisfied for each rule is known. If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one value that represents the result of the antecedent for that rule.

The implication function then modifies that output fuzzy set to the degree specified by the antecedent. Since decisions are based on the testing of all of the rules, the results from each rule must be combined in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. The input for the defuzzification process is the aggregated output fuzzy set and the output is a single value. This is summarized in the figure 62.

Appendix B Hardware and Software environment of development

Hardware overview

Imote2 Processor Board

The Crossbow Imote2 is an advanced sensor network node platform designed for demanding wireless sensor network applications requiring high CPU/DSP and wireless link performance and reliability.

Figure 65. Imote2

The platform is built around Intel's XScale® processor, PXA271. It integrates a ZigBee / 802.15.4 radio (TI CC2420) with an on-board antenna.

Figure 66. Imote2 sensor board

It exposes a "basic sensor board" interface, consisting of two connectors on one side of the board, and an "advanced sensor board" interface, consisting of two high density connectors on the other side of the board. The Imote2 is a modular stackable platform and can be stacked with sensor boards to customize the system to a specific application, along with a "battery board" to supply power to the system.

ITS400 Sensor Board

The basic sensor board is designed to connect to the basic connectors on the Imote2. It contains a 3d Accelerometer, advanced temp/humidity sensor, light sensor and 4 channels A/D. It is a pass through board to allow stacking with another sensor/communication board.

The sensor board is multi-sensor board that combines a popular set of sensors for wireless sensor network applications, including:

- ST Micro LIS3L02DQ 3d 12 bit ±2g accelerometer
- High Accuracy, ±0.3°C Sensirion SHT15 temperature/humidity sensor
- TAOS TSL2651Light Sensor
- Maxim MAX1363 4 Channel General Purpose A/D for quick prototyping
- TI Tmp175 Digital Temperature Sensor with two-wire interface

Figure 67. ITS400 Sensor board

Software overview

Imote2.Builder SDK

Crossbow Imote2.Builder SDK is an add-on to Microsoft Visual Studio 2005 and .NET Micro Framework 2.0. It provides a development environment for wireless sensor networking (WSN) applications leveraging the Microsoft .NET Framework and makes it easy for developers working in Microsoft Visual Studio, to experiment with wireless sensor applications. Moreover, wireless sensor applications can be rapidly prototyped, with proof of concept in hours or days, not weeks or months.

Microsoft .NET Micro Framework

The .NET Micro Framework is a new execution model for extremely small, resource-constrained devices.

It brings the advantages of .NET—the security and reliability of managed code, the ease of development in Visual Studio, and the power of the C# language and the .NET libraries—to a smaller class of device than Microsoft has ever targeted before.

The architecture of the .NET Micro Framework provides developing the firmware as managed C# applications. It does this by making use of specialized versions of the .NET common language runtime (CLR) and the .NET Framework class library.

The .NET Micro Framework architecture is optimized for small devices and makes it possible for you to use Microsoft Visual Studio® and C# to program an embedded system. These devices include, but are not limited to: Sensor networks, Robotics, GPS navigation, Wearable devices, Medical instrumentation, Industrial automation devices and other small devices that require an efficient, low-resource-consuming Microsoft .NET client

The .NET Micro Framework is available for an increasing variety of hardware. Its architecture is designed to be extremely flexible and highly adaptable to new hardware platforms. .NET Micro Framework architecture is divided into the four distinct layers shown in figure 68.

The hardware layer contains the microprocessor and other circuitry that you select. The .NET Micro Framework runs on a growing number of hardware platforms. It can also run on top of operating systems such as Microsoft Windows® and some other proprietary operating systems belonging to companies partnering with Microsoft. It is possible to port the .NET Micro Framework onto other chipsets. The runtime component layer (RCL) includes three components: the .NET Micro Framework common language runtime (CLR), a hardware abstraction layer (HAL), and a platform abstraction layer (PAL).

The .NET Micro Framework CLR is a subset of the .NET Framework CLR, which is the run-time environment provided by the .NET Framework. The .NET Micro Framework CLR provides robust application support. It manages memory, thread execution, code execution, and other system services. The CLR is able to provide all of these features and services from a very small memory footprint. It occupies only about 390 KB of memory when all of the provided functionality is being used. This memory-usage estimate is for Microsoft's implementation framework for existing platforms.

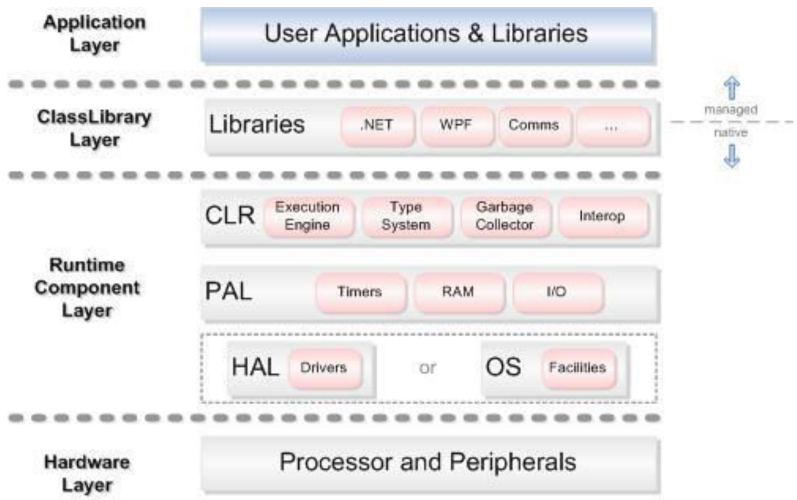


Figure 68. .NET Micro Framework architecture

The CLR is fast; it performs at about 15,000 managed method calls per second at 27.6 MHz. (These results are based on the average method-call complexity developed for the SPOT watch application.) Currently, the CLR supports only C# as its programming language. Features of the .NET Micro Framework CLR include:

- Numeric types, class types, value types, arrays (one-dimensional only), delegates, events, references, and weak references
- Synchronization, threads, and timers
- Reflection
- Serialization
- Garbage collection
- Exception handling
- Time classes, including DateTime and TimeSpan
- Time-sliced thread management

Exceptions to and extensions of the CLR include:

- Execution constraints that limit call durations and prevent failures.
- Strings represented internally as UTF-8 and exposed as Unicode.
- No support for multidimensional (sparse or jagged) arrays.
- A global, shared string table for well-known values (types, methods, and fields) that reduces RAM and ROM and reduces the number of cross-references among assemblies.
- No virtual tables for method resolution, which saves RAM.
- A WeakDelegate class to handle dangling references to delegates.
- Support for extending the hardware chipset by programming its drivers directly in C#, using the Managed Drivers Framework. Hardware devices compatible with industry communication standards — such as GPIO, serial, SPI, or I^2C — can be added to the chipset and used by a managed C# application.

The HAL and the PAL control the underlying system hardware. Both the HAL and the PAL are groups of C++ functions called by the CLR. The PAL functions are independent of the hardware and should not need to be ported. However, you must write your own version of the HAL for your hardware platform.

The bootstrap code is an additional piece of code that is associated with the HAL. The bootstrap code initializes the low-level hardware when the device is turned on. It then starts the CLR, which performs the higher-level initializations. The bootstrap code performs its tasks through calls to the HAL and assembly-language routines. Other than starting the CLR, it has no interaction with the code preceding it in the software architecture.

The class library included with the .NET Micro Framework is an object-oriented collection of reusable types that you can use to develop embedded applications. It includes C# libraries that provide support for the following:

- Encryption
- Debug, graphics, and shell DLLs
- The CLR API class library and the CLR core lib
- Access for managed C# applications to extended chipsets that support specific communications standards, such as GPIO, serial, SPI, or I^2C

The top layer of the .NET Micro Framework contains managed applications that you create for your devices. The types of applications you develop depend entirely on the hardware in the device.

List of my publications

Papers

1- S. Nourizadeh, Y.Q. Song, J.P. Thomesse, "CodaQ – A Context-aware and QoS-aware Middleware for Activity Monitoring", ICOST2011, Montreal, Québec

2- S. Nourizadeh, C. Deroussent, "Maisons Vill'Âge®: Smart Use of Home Automation Sensor and Actuator Networks for Healthy Aging", ICOST2011, Montreal, Québec

3- S. Nourizadeh, Y.Q. Song, J.P. Thomesse, C. Deroussent, "A Context-aware Tele-homecare System for Senior Citizens", MedeTel 2010, Luxembourg

4- S. Nourizadeh, Y.Q. Song, J.P. Thomesse, "An Adaptive Hierarchical Routing Protocol for Wireless Ad-hoc Sensor Networks" IEEE NGMAST'09 2009, UK

5- S. Nourizadeh, Y.Q. Song, J.P. Thomesse, "A Failure Adapted, Load-balanced Distributed Routing for Wireless Ad-hoc Sensor Networks - a Fuzzy Logic based approach" IEEE EUC-2009 Vancouver, CA

6- S. Nourizadeh, Y.Q. Song, J.P. Thomesse, "A Distributed Fuzzy-based Failure Management for Wireless Sensor Networks" CTRQ2009, Colmar - France

7- S. Nourizadeh, C. Deroussent, Y.Q. Song, J.P. Thomesse, "Medical and Home automation Sensor Networks for Senior Citizens Telehomecare", IEEE ICC MAN'2009 Dresden - Germany

8- S. Nourizadeh, C. Deroussent, Y.Q. Song, J.P. Thomesse, "A Distributed Elderly Healthcare System", MobiHealth2009, Porto – Portugal

9- S. Nourizadeh, Y.Q. Song, J.P. Thomesse, X. Seoulchre, "Un Système Distribué de Télémédecine pour Les Seniors" SFTAG09, Troy, France

10- S. Nourizadeh, Y.Q. Song, J.P. Thomesse, "A Location-Unaware Distributed Clustering Algorithm for Mobile Wireless Sensor Networks Using Fuzzy Logic", FET2007 Toulouse – France

Posters

1- S. Nourizadeh, Y.Q. Song, J.P. Thomesse, "Wireless Sensor Networks for Ambient Assisted Living" SensorNets2009, Monastir - Tunisie

2- S. Nourizadeh, Y.Q. Song, J.P. Thomesse, "A Novel Adaptive Routing Protocol for Ad-hoc Wireless Sensor Networks" SensorNets2009, Monastir - Tunisie

3- S. Nourizadeh, C. Deroussent, Y.Q. Song, J.P. Thomesse,"Medical and Home automation Sensor Networks for Senior Citizens Telehomecare", FFSRT Telemedmeeting 2009, Nov. 2009, Archamps – Franc.

Nancy-Université
INPL

AUTORISATION DE SOUTENANCE DE THESE
DU DOCTORAT DE L'INSTITUT NATIONAL
POLYTECHNIQUE DE LORRAINE

oOo

VU LES RAPPORTS ETABLIS PAR :
Monsieur Norbert NOURY, Professeur, Universités UCBL, INL, Site INSA, VILLEURBANNE
Monsieur Bessam ABDULRAZAK, Professeur, Université de Sherbrooke (UdeS), Canada

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

Monsieur NOURIZADEH Shahram

à soutenir devant un jury de l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE, une thèse intitulée :

"Un système de télésanté contextuel avec support de qualité de service pour le maintien à domicile (A context-aware and QoS-aware telehomecare system). "

en vue de l'obtention du titre de :

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Spécialité : « **Informatique** »

Fait à Vandoeuvre, le 30 juin 2011
Le Président de l'I.N.P.L.,
F. LAURENT

INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE — NANCY

Institut National Polytechnique de Lorraine
inpl@inpl-nancy.fr | www.inpl-nancy.fr

2 av. de la Forêt de Haye - BP 3
54501 Vandoeuvre-lès-Nancy Cedex
FRANCE
Tél. : +33(0)3 83 59 59 59
Fax : +33(0)3 83 59 59 55