SECURED WSN-INTEGRATED CLOUD COMPUTING FOR U-LIFE CARE (SC^3)

by

SC^3 Team[1]

Technical Report

Ubiquitous Computing Lab
Kyung Hee University

July 2009

Approved by _____
Project Supervisor

_____

Date _____

[1] Le Xuan Hung (team leader), Phan Tran Ho Truc, La The Vinh, Asad Masood Khattak, Manhyung Han, Mohammad M. Hassan, Dang Viet Hung, Sungyoung Lee, Young-Koo Lee.

UBIQUITOUS COMPUTING LAB

KYUNG HEE UNIVERSITY

ABSTRACT

SECURED WSN-INTEGRATED CLOUD COMPUTING FOR U-LIFE CARE

by SC^3 Team

Project Supervision Professor Sungyoung Lee
Department of Computer Engineering

Ubiquitous Life Care (u-Life care) nowadays becomes more attractive to computer science researchers due to a demand on a high quality and low cost of care services at anytime and anywhere. Many works exploit sensor networks to monitor patient's health status and movements to provide care services to them. It requires sensory data to be quickly transmitted and processed so that physicians, clinics, and other caregivers can access conveniently via Internet. Most existing life care systems rely on their own data center to store and process sensory data. It brings a high cost to maintain the system, yet the performance is not reliable and a limited number of services can be provided. This paper presents our study and development of a Secured Wireless Sensor Network (WSN) - integrated Cloud Computing for u-Life Care (SC^3). We abstract SC^3 in three levels: secure wireless sensor networks (WSNs), secure Clouds, and secure end-users. WSNs can be deployed in home or office environments. Sensor data is collected and sent to the Clouds. In the Cloud, an Activity Recognition and Ontology Engine are deployed to process raw sensory data, infer high lever activities, and make response decision. Different users, committees such as hospitals, clinics, researchers, or even patients themselves can access their data in the Clouds. SC^3 is composed of five major modules: security for WSNs (trust management), security and privacy control for Clouds (authentication and access control), integration mechanism of wireless sensor networks to Clouds, sensor data dissemination mechanism, and a dynamic collaboration mechanism between different Cloud providers (CLPs). To enable u-Life care, we also introduce a new method of Activity Recognition and Ontology Engine. An early result of our development is also presented. Our proposed SC3 can help in enhancing capabilities and provides tremendous value by achieving efficient use of software and hardware investments. Our infrastructure drives profitability by improving resources utilization and increasing their scalability while maintaining strong privacy and security essential in u-Life care. SC^3 can provide cost efficient model for automating hospitals and other life care agencies, managing real-time data from various sensors, efficiently disseminating information to consumers, support privacy and strong authentication mechanism, reducing IT complexity and at the same time introducing innovative solutions and updates. Our versatile architecture makes it possible to launch web 2.0 applications quickly and also upgrade u-life care IT applications easily as and when required. Our automated secure framework of cloud computing would provide increasingly cheaper and innovative services.

Table of Contents

Table of Figures

Chapter 1

INTRODUCTION TO CLOUD COMPUTING

1 Introduction

1.1 What is Cloud Computing?

The Cloud computing, coined in late of 2007, currently emerges as a hot topic due to its abilities to offer flexible dynamic IT infrastructures, QoS guaranteed computing environments and configurable software services. Cloud computing can be defined as follows:

> "A Cloud is a type of parallel and distributed system consisting of a *collection of interconnected and virtualized computers* that are *dynamically provisioned* and presented as one or more *unified computing resources* based on service-level agreements established through negotiation between the service provider and customers and can be ubiquitously accessed from any connected devices over the internet"

Cloud computing started quietly from several seeding technologies such as grid computing, virtualization, SalesForce.com innovative subscription-based business model or Amazon's effort to scale their e-commerce platform. However, it differs from traditional ones in that: (1) it is **massively scalable**, (2) can be encapsulated as an **abstract entity** that delivers different levels of services to customers anywhere, anytime, and (3) it is driven by economies of scale that is the services can be dynamically configured (via virtualization or other approaches) and delivered "**on-demand**".

The Web search popularity, as measured by the Google search trends during the last 12 months, for terms "Cluster computing", "Grid computing", and "Cloud computing" is shown in Figure 1. From the Google trends, it can be observed that cluster computing was a popular term during 1990s, from early 2000 Grid computing become popular, and recently Cloud computing started gaining popularity. Meanwhile, market-research firm IDC expects IT Cloud-services spending to grow from about $16 billion in 2008 to about $42 billion by 2012 as Figure 2 shows. IDC also predicts Cloud computing spending will account for 25 percent of annual IT expenditure growth by 2012 and nearly a third of the growth the following year.

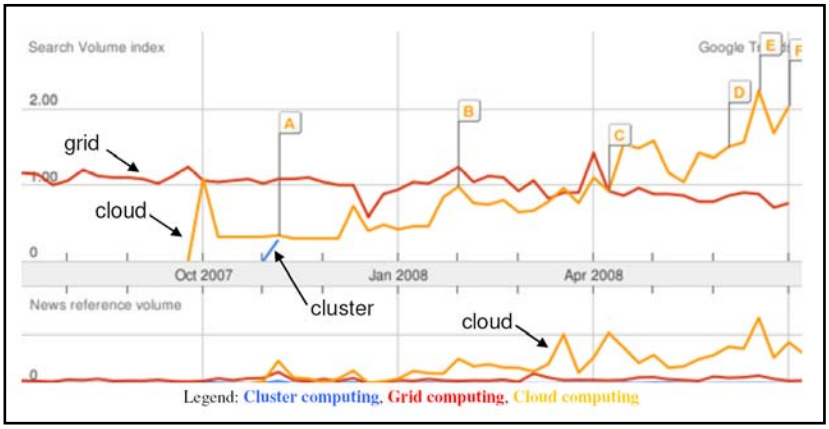


Figure 1 Cloud computing in Google Trends

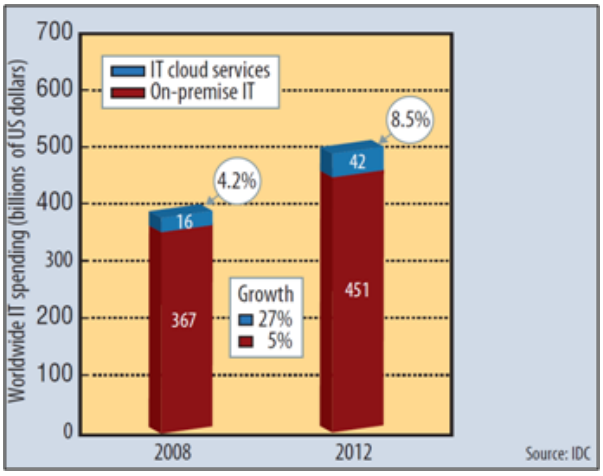


Figure 2 IT service investment

Cloud Computing has many benefits that the public sector and government IT organizations are certain to want to take advantage of. In very brief summary form they are as follows:

Reduced cost, higher gains: Cloud technology is paid incrementally, saving organizations money.

Increased storage: Organizations can store more data than on private computer systems.

Highly automated: No longer do IT personnel need to worry about keeping software up to date.

Flexibility: Cloud computing offers much more flexibility than past computing methods.

More mobility: Employees can access information wherever they are, rather than having to remain at their desks.

Allows IT to shift focus: No longer having to worry about constant server updates and other computing issues, government organizations will be free to concentrate on

innovation.

In Cloud computing, customers do not own the infrastructure they are using; they basically rent it, or pay as they use it. One of the major selling points of cloud computing is **lower costs**. Companies will have lower technology-based capital expenditures, which should enable companies to focus their money on delivering the goods and services that they specialize in. There will be more device and location independence, enabling users to access systems no matter where they are located or what kind of device they are using. The sharing of costs and resources amongst so many users will also allow for efficiencies and cost savings around things like performance, load balancing, and even locations (locating data centers and infrastructure in areas with lower real estate costs, for example). The general architecture of Cloud computing is shown in Figure 3.

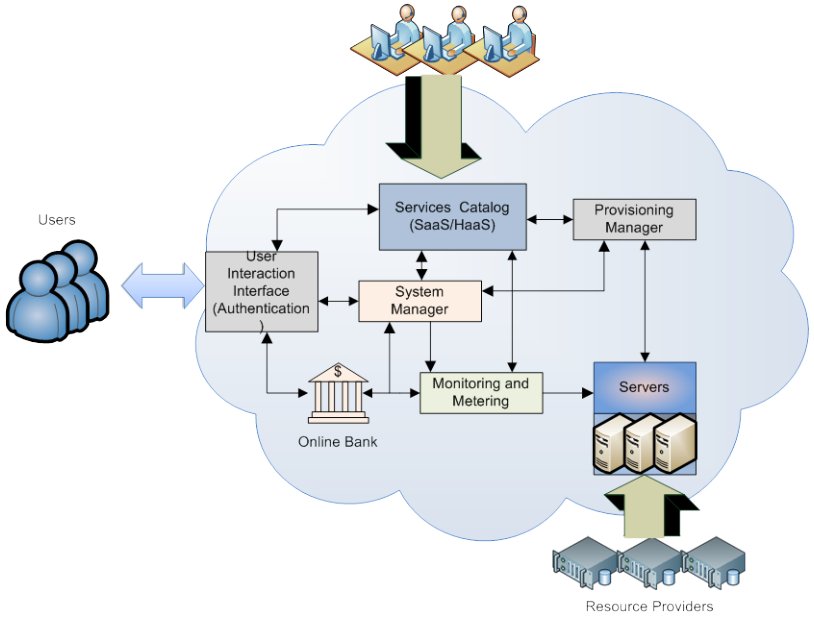


Figure 3 General Architecture of Cloud Computing

1.2 Why Cloud Computing in u-Life care?

As the standard of living rises, people are more interested in their health and desire well-being life. Today due to aging of population, rising cost of workforce and high quality treatment, threat of new panepidemies and diseases, the cost of life care or healthcare system is increasing worldwide. Accoridng to OECD (Organizaiton of Economic Cooperation and Development) Health data 2008 (shown in Figure 4), total health spending accounted for 15.3% of GDP in the United States in 2006, the highest share in the OECD, and more than six percentage points higher than the average of 8.9% in OECD countries. Korea was 6.4% of GDP to health in 2006. The United States also ranks far ahead of other OECD countries in terms of total health

spending per capita, with spending of 6,714 USD (adjusted for purchasing power parity (PPP)), more than twice the OECD average of 2,824 USD in 2006. For Korea it was 1480 USD.

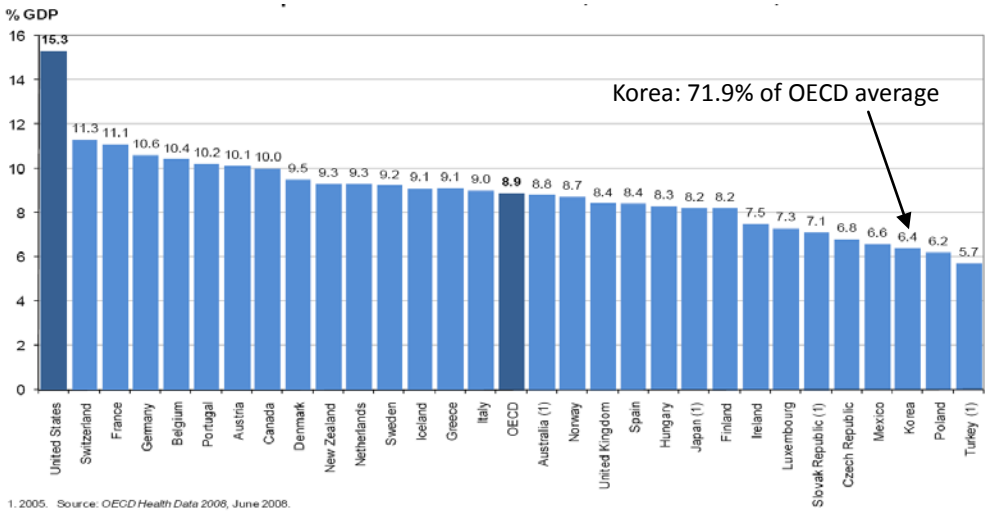


Figure 4 Health expenditure as a share of GDP, OECD countries, 2006

In Korea**,** the number of aging population is increasing as shown in Figure 5. The national health expenditure of Korea as compare to GDP is shown in Figure 6. We can see from that the financial crisis in healthcare is increasing and Governments are trying to increase the budget every year.

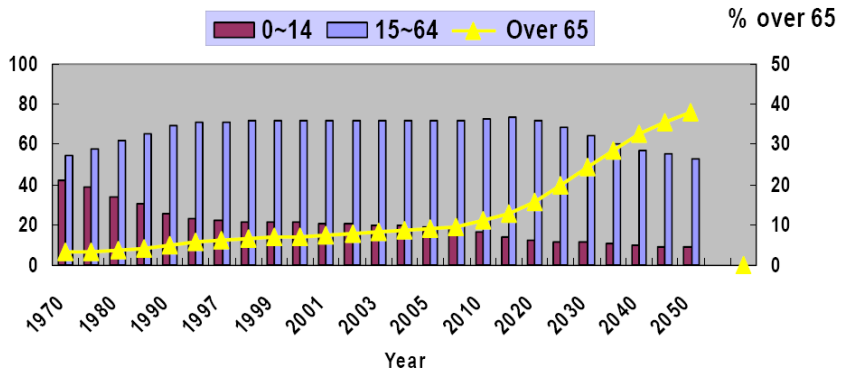


Figure 5 Population by Age group in Korea

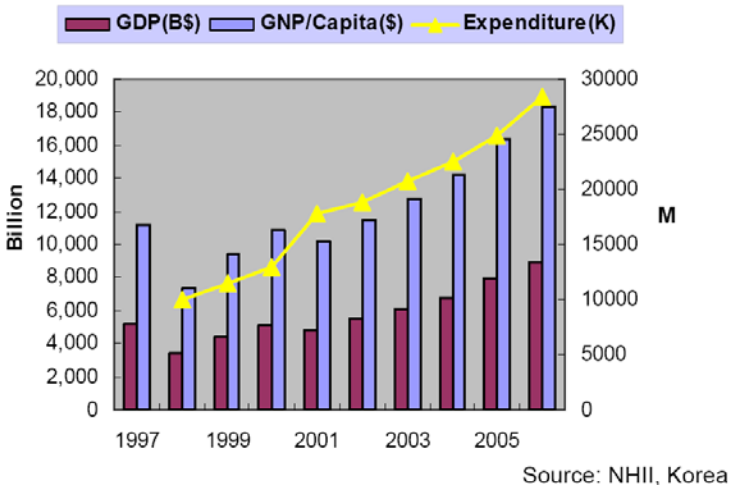


Figure 6 National Healthcare expenditure compare to GDP

To maintain the quality and availability level of life care services with minimum cost, Cloud computing can provide a **powerful, flexible, and cost-effective infrastructure** for life care services that can fulfill the vision of "ubiquitous life care" that is providing life care to people anywhere at any time while increasing both the coverage and the quality. Because of its elasticity, scalability, pay-as-you-go model, Cloud computing can potentially provide **huge cost savings, flexible high-throughput, and ease of use** for life care services. For example, with life care providers looking at automating processes at lower cost and higher gains, Cloud computing can act as an ideal platform in the Life care IT space. Hospitals could share Cloud computing infrastructure with vast number of systems linked together for reducing cost and increasing efficiency. Patient information and data can be accessed globally and resources can be shared by a group of hospitals rather than each hospital having a separate IT infrastructure. Cloud Computing would help hospitals to achieve more efficient use of their hardware and software investments and increase profitability by improving the utilization of resources to the maximum. By pooling the various Life care IT resources into large clouds, hospitals can reduce the cost and increase utilization as the resources are delivered only, when they are required. The use of cloud computing architecture helps is in eliminating the time and effort needed to roll a Life care IT application in a hospital. Integrating with wireless sensor networks, the cloud can provide real time information of the environment for collaboration and knowledge sharing in data intensive research and analysis, especially in the health and biomedical arena. The Cloud could, for instance, provide a flexible platform for public-health departments to upload real time health data in a timely manner to assist state and national health officials in the early identification and tracking of disease outbreaks, environmental-related health problems, and other issues.

1.3 Problems of Existing Cloud computing to support u-Life care

❖ ***Poor Security and Privacy Support***

Data for life care services normally composes of personal information, contextual information (e.g. location, user activity information), medical data (e.g. medical history, drug information, medical health record), etc. Such information is **highly sensitive** and people do not want to disclose them to the public. For example, a patient with HIV positive test may not want to expose his result to the other, even to their family.

Storing data in Cloud leads to **more security and privacy problems than traditional computing** systems such as distributed systems or grid computing systems. Sensitive data processed outside the enterprise brings with it an **inherent level of risk**, because outsourced services bypass the "physical, logical and personnel controls" IT shops exert over in-house programs. More dangers and vulnerabilities may cause disrupts of services, theft of information, loss of privacy, damage of information. On the other hand, because any one can access to Clouds, it brings more chances for malicious users to launch their hostile programs. Hostile people can also give instructions to good programs, or bad guys corrupting or eavesdropping on communications.

❖ ***No Existing Infrastructure for Integration of WSN to Cloud***

In the past few years, wireless sensor networks (WSNs) have been gaining increasing attention to create decision making capabilities and alert mechanisms, in many Life care application areas including Life care monitoring for patients, environmental monitoring, pollution control, disaster recovery, military surveillance etc. For example, MIT wireless sensor ring as shown in Figure 7can measure heart rate, heart rate variability, Oxygen saturation and blood pressure for the person wearing the ring.

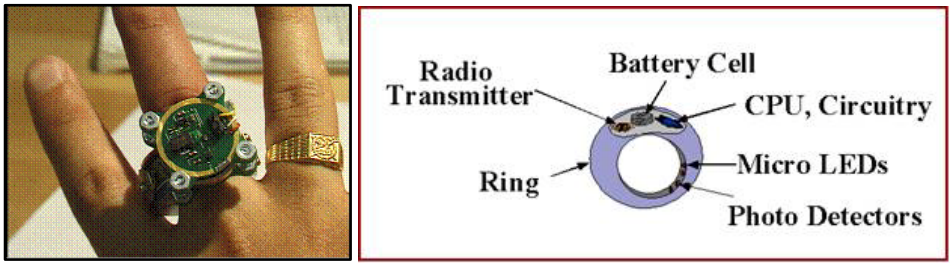


Figure 7 MIT wireless sensor ring and its internal architecture

Collection, analysis (knowledge processing, ontology reasoning etc.), storing and disseminating of these sensor data is a great challenge since sensor nodes constituting a WSN have limited sensing capability, processing power, and communication bandwidth. There is a lack of uniform operations and standard

representation for sensor data.

Cloud computing provides a powerful, flexible, cost-effective and massively scalable platform that enables real-time sensor data collection and the sharing of huge computational and storage resources for sensor data processing and management Using this collection of real-time sensor derived data to various Cloud computing applications, we can have a significant transformation in our ability to "see" ourselves and our planet. For example, in case of health care, integrating WSN to Cloud computing can provide real-time information of the environment for collaboration, analysis and knowledge-sharing of the early identification and tracking of disease outbreaks, environmental-related health problems, and other issues

- ***Lacking of Efficient Information Dissemination Mechanism***

To deliver published sensor data or events to appropriate users or applications from Sensor-Cloud, there is a need of an information dissemination mechanism that matches published events with subscriptions efficiently. Designing an efficient content or event matching algorithm is a key especially for the range predicate or overlapping predicate case. It is difficult to construct an effective index for multidimensional range predicates. It is even more challenging if these predicates are highly overlapping. In u-Life care application scenario, doctors and caregivers may express their interests into a range (*i.e*. 35 < *body temperature* < 37)). As patient states are changed continuously within a certain range (min, max) like body temperature and heartbeat, each classified consumer will require events in various ranges to safely take care of patients. Also there are other issues to be considered in the design of information dissemination system like maintaining the flexible expressiveness of predicates that is the ability to provide powerful subscription schema capable to capture information about events, guarantees the scalability with respect to the number of subscribers and the published events and supporting system adaptability.

- ***No Support of Dynamic Collaboration between Cloud Providers in case of Service Level Agreement Violation (SLA)***

As consumers of different Cloud applications rely on Cloud Providers (CLP) to supply all their computing needs (process, store and analyze huge sensor data and user generated data) on demand, they will require specific QoS to be maintained by their providers in order to meet their objectives and sustain their operations. Existing commercial Cloud services are proprietary in nature. They are owned and operated by individual companies. Each of them has created its own closed network, which is expensive to setup and maintain. So, if the CLP is unable to provide quality of service to the end-user requests (in the case when huge sensor data needs to process for critical U-Life care scenario), it may result in *Service Level Agreement* (SLA) violation and end up costing the provider. So there is a need of a dynamic collaboration between Cloud providers. But choosing the best combination of Cloud providers for

dynamic collaboration is the major challenge in terms of cost, conflicts between providers, time and QoS.

1.4 Practical Usage

Our proposed SC^3 can be deployed for various u-Life care services, including but not limited to:

- **Safety monitoring services for home users**
  - SC^3's WSN can monitor home user's movement, location by using various sensors. The sensor data is then disseminated to the Clouds, from that SC^3's Life care services such as emergency caregivers can monitor and has immediate response in case of emergent situations like heart attack.

- **Information sharing services**
  - With SC^3, patient information and data can be accessed globally and resources can be shared by a group of hospitals rather than each hospital having a separate IT infrastructure. Cloud computing would help hospitals to achieve more efficient use of their hardware and software investments and increase profitability by improving the utilization of resources to the maximum
  - The SC^3 can provide a flexible platform for public-health departments to upload real-time health data in a timely manner to assist state and national health officials in the early identification and tracking of disease outbreaks, environmental-related health problems, and other issues.

- **Emergency-connection services**
  - SC^3 can be deployed to real-time monitor home environments, including gas, fire, thief, etc. Through SC^3, an alarm system connects to users, u-119, police department can give an emergency alert in case any emergent situation occurs.

- **Users can monitor their home, their family health anywhere, any time with any device**
  - SC^3 Clouds and WSN enable user to access their home environment, their family's health information with any kind of connected devices over Internets such as cell phone, PDA, laptop, computer.

1.5 Contribution of this Study

Our proposed SC^3 can help in enhancing capabilities and provides tremendous value

by achieving efficient use of software and hardware investments. Our infrastructure drives profitability by improving resources utilization and increasing their scalability while maintaining strong privacy and security essential in u-Life care. SC^3 can provide cost efficient model for automating hospitals and other life care agencies, managing real-time data from various sensors, efficiently disseminating information to consumers, support privacy and strong authentication mechanism, reducing IT complexity and at the same time introducing innovative solutions and updates. Our versatile architecture makes it possible to launch web 2.0 applications quickly and also upgrade u-life care IT applications easily as and when required. Our automated secure framework of cloud computing would provide increasingly cheaper and innovative services.

Technically, our SC^3 infrastructure can contribute in the following ways in u-Life care:

- ✓ Our architecture helps in eliminating the time and effort needed to roll a heal thcare IT application in a life care centre
- ✓ Flexible and swift access to expert opinion
- ✓ Intelligent personal health monitoring system
- ✓ Synergy of information from individual sensors (better insight into the physiol ogical state and level of activity).
- ✓ Hospitals, silver care centers and life care agencies could share our secured infrastructure with vast number of systems linked together (i.e. secured sensor network to support real time information) for reducing cost and increasing efficiency. This means real-time availability of patient information for doctors, nursing staff and other support services not within the country but possibly across various countries as medical professionals can access patient information from any internet enabled device without installing any software.
- ✓ The EMR software or the LIS software and information can be located in our Cloud and not on the users or computer. Patient information and data can be accessed globally maintaining proper privacy and security policy and resources can be shared by a group of hospitals or life care agencies rather than each hospital having a separate IT infrastructure.
- ✓ Rapid response to critical life care regardless of geographic barriers (anytime, anywhere)
- ✓ Management of medical expertise also in rural areas
- ✓ Swift medical care in emergencies and medical data management in

catastrophes

- ✓ Promotion of healthy life styles with continuous health monitoring
- ✓ Savings for ubiquitous healthcare service providers and patients in procedural, travel, and claim processing cost
- ✓ Reduced use of traditional emergency services
- ✓ Improved non-emergency services
- ✓ Decreased waiting time for nonemergency services
- ✓ Greater awareness of services among rural and remote residents and caregivers
- ✓ Timely accessibility of critical information in the event of emergencies.

Chapter 2

RELATED WORK

2 Related Work

Some existing Cloud computing efforts are as follows:

- IBM Introduces 'Blue Cloud' Computing, CIO Today - Nov 15 2007
- IBM, EU Launch RESERVOIR Research Initiative for Cloud Computing, IT News Online - Feb 7, 2008
- Google and Salesforce.com in Cloud computing deal, Siliconrepublic.com - Apr 14 2008
- Demystifying Cloud Computing, Intelligent Enterprise - Jun 11 2008
- Yahoo realigns to support Cloud computing, 'core strategies', San Antonio Business Journal - Jun 27 2008
- Merrill Lynch Estimates "Cloud Computing" To Be $100 Billion Market, SYS-CON Media - Jul 8

Now we will describe some existing work related to u-Life care.

2.1 Korea u-Care System for a Solitary Senior Citizen (SSC)

Numerous projects within industry and academia are already started or being used in reality. in 17 July 2008, an RFP of Ministry for Health, Welfare and Family Affairs, Korea has released **u-Care System for a Solitary Senior Citizen** (SSC) [1].

The system provides a number of featured services including:

24 hours × 365 days safety monitoring services for a SSC: monitoring SSC's movement or location by using motion or tripwire sensors, voice communication between a SSC and a life-care giver at all times, emergency call service for heart attack, etc.
Emergency-connection services: real-time monitoring of gas spillage or fire by using gas and detection sensors, if any emergency situation occurs like gas or fire, automatically notify to u-119 system.
Information sharing services: sharing a SSC database to related organization such as fire-station, care givers, or central information system, and a history of care givers activities to build an efficient SSC management systems, etc.

In this system, *u-Care Center* maintains Data Server, DB Server, and Web Server to store data from SSC's home and provide access to various services, as illustrated in Figure 8. However, its biggest issue is a **high cost of deployment** when it becomes widely used. With recent advanced in MEMS technology which enables low cost

sensors, each user may carry on a number of sensors and wearable devices, leading to a huge amount of data produced. On the other hand, more and more life care applications and services access to the system to provide better care for users. It obviously causes a problem of economy to maintain the system.

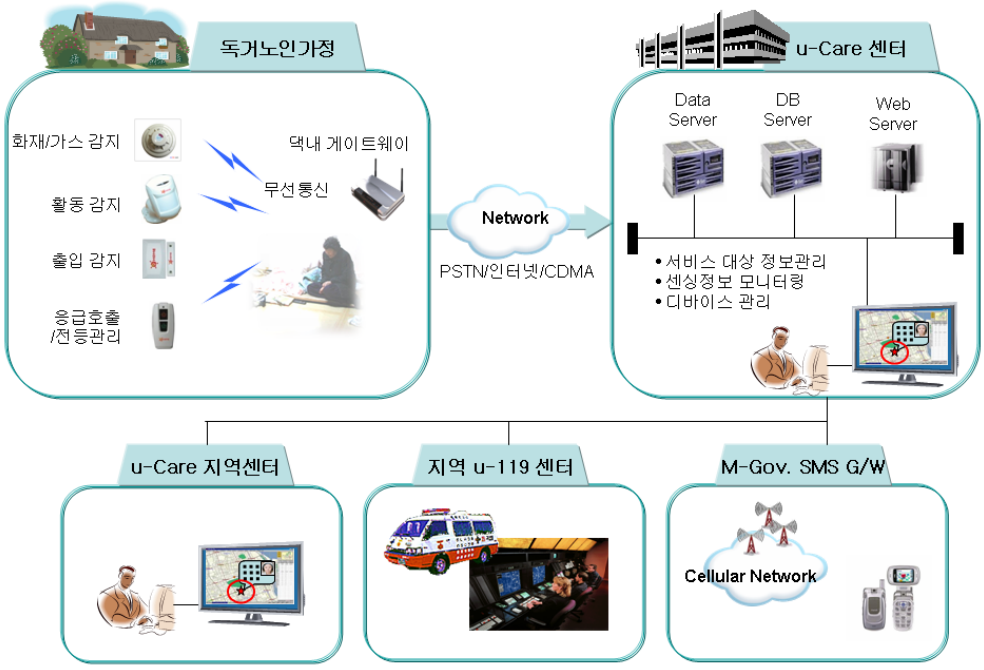


Figure 8 Hardware architecture of u-Care System for a Solitary Senior Citizen (source: Korea Ministry)

In another sector of healthcare applications, several providers explore advantages of Cloud computing to reduce cost and increase care services. Among those, Microsoft and Google are two pioneers who bring Cloud healthcare platforms to reality.

2.2 Microsoft HealthVault

Microsoft developed a platform to store and maintain health and fitness information, called **HealthVault** [1]. It is a cloud service that helps people collect, store, and share their personal health information. HealthVault's data can come from providers, pharmacies, plans, government, employers, labs, equipment and devices, and from consumers themselves. Access to a record is through a HealthVault account, which may be authorized to access records for multiple individuals, so that a mother may manage records for each of her children or a son may have access to his father's record to help the father deal with medical issues.

2.3 Google Health

Meanwhile, Google provides a personal health information centralization services,

known as **Google Health** [3]. The service allows Google users to volunteer their health records, either manually or by logging into their accounts at partnered health services providers, into the Google Health system, thereby merging potentially separate health records into one centralized Google Health profile. Volunteered information can include health conditions, medications, allergies, and lab results. Once entered, Google Health uses the information to provide the user with a merged health record, information on conditions, and possible interactions between drugs, conditions, and allergies.

In general, HealthVault and Google Health serve as Cloud health information storages and operate separately. As consumers of different Cloud applications rely on Cloud Providers (CLP) to supply all their computing needs (process, store and analyze huge sensor data and user generated data) on demand, they will require specific QoS to be maintained by their providers in order to meet their objectives and sustain their operations. To solve **the problem of Cloud interoperation**, an Unified Cloud Interface (UCI) standardization has been proposed.

2.4 Amazon's Cloud computing based Healthcare efforts

At an invitation-only event sponsored by Harvard Medical School and Amazon Web Services, a few dozen experts convened in Boston for a day to ponder the possibilities of cloud computing in their work. Participants included health care IT leaders, academics, biomedical researchers, medical and scientific consulting firm representatives, and officials from vendors like Amazon, Oracle, and Hewlett-Packard [54].

For its part, Amazon in recent weeks unveiled the AWS Hosted Public Data Sets, or "Public Data Computing Initiative," which provides on the cloud a "hosted-for-free, centralized public repository" for data -- such as United States census and human genome research data -- useful to researchers,

Now we will describe some existing solutions or standard regarding Cloud computing.

2.5 Secured Cloud Overlay: VPN-Cubed

One of the security challenges of Cloud Computing - and specifically Infrastruc ture as a Service (IaaS) is securely connecting enterprise network to one or m ore Cloud providers without deploying VPN hardware.

VPN-Cubed™ is the first commercial solution by CohesiveFT that enables customer control in a cloud, across multiple clouds, and between private infrastructure and the clouds. VPN-Cubed provides an overlay network that allows the user to control of addressing, topology, protocols, and encrypted communications for devices deployed to virtual infrastructure or cloud computing centers [55]. When using

public clouds corporate assets are going into 3rd party controlled infrastructure. VPN-Cubed gives you flexibility with control in 3rd party environments. Here consumer has most of the responsibility to secure his/her own data.

2.6 Unified Cloud Interface (UCI) standardization

The **Unified Cloud Interface** (UCI) standardization [4] or Cloud broker will serve as a common interface for the interaction with remote platforms, systems, networks, data, identity, applications and services. UCI will be composed of a semantic specification and an ontology. The ontology provides the actual model descriptions, while the specification defines the details for integration with other management models. One of the key drivers of the unified cloud interface is to create an API about other API's. A singular programmatic point of contact that can encompass the entire infrastructure stack as well as emerging cloud centric technologies all through a unified interface. The draft proposal will be submitted for approval by the Internet Engineering Task Force (IETF) for inclusion as a XMPP Extension and presented at the IEEE International Workshop on Cloud Computing (Cloud 2009) being held in May 18-21, 2009, in Shanghai, China.

2.7 Problems of Existing Works

- ***Weak security support***

  - First, the customer needs to know her data is encrypted so nosey sysadmins at the cloud data center can't troll through the data for interesting tidbits. If the information is encrypted, who controls the encryption/decryption keys, the customer or the cloud vendor?
  - Integrity relates to the integrity of the data, in that it changes only in response to duly authorized transactions. So we need standards to ensure that. But they don't exist -- yet.
  - The last nagging security issue is availability: Will the data be there whenever you need it? The answer here is an unqualified "maybe." In February of this year, Amazon's S3 went down for almost four hours, wreaking havoc on several companies that use and depend on the S3 Cloud. Amazon ascribed the cause to an unexpected spike in customer transactions.

- ***Weak Privacy Support***

  - Private health data can goes public by mistake: Part of consumers' reticence to sign up for electronic personal health-care records — with or without services "in the cloud" — has to do with a handful of recent high-profile data breaches. In April, the largest health insurer in the U.S., WellPoint, disclosed that records on as many as 130,000 of its customers had leaked out and become publicly available over the Internet. User heath data and information uploaded into Clouds are not controlled by user
  - Consumer's privacy may get loss in the cloud: Is there a law that keeps your data from being misused? Yes. It is Health Insurance Portability and Accountability

Act (HIPAA), but it does not offer health-care service themselves. Right now, disclosure of health information is out of control.

❖ ***No infrastructure to support WSN integration to Cloud***

- The existing Cloud based Healthcare system does not integrate wireless sensor network which is necessary to get real time information of patient or environment to monitor and analysis emergency situation.
- Appropriate information dissemination mechanism is not explicitly addressed in the existing system to deliver sensor data or events to appropriate users of Cloud applications who subscribed. Also there is a need to match published events with subscriptions efficiently. A fast, scalable and efficient event matching algorithm is required for information dissemination system on Sensor-Cloud framework.

❖ ***Lack of Dynamic Collaboration between Cloud providers***

- When the Cloud provider is unable to provide quality of service to the end-user requests, it may result in service–level agreement (SLA) violation and end up costing the provider
- Existing commercial Cloud services are proprietary in nature. They are owned and operated by individual companies. Each of them has created its own closed network, which is expensive to setup and maintain. Existing Cloud based solution does not consider the dynamic collaboration between Cloud providers which is obvious in near future.

Table 1 shows a comparison of the above existing work.

Table 1. A Comparison of Existing Works

Existing Works / Features	Korea u-Care System	MS. HealthVault	Google Health	Amazon	VPN-Cube™	UCI
Security	X	Weak	Weak	Weak	Weak	X
Privacy Control	Weak	Weak	Weak	Weak	Weak	X
USN Integration to Cloud	X	X	X	X	X	X
Sensor data dissemination to Cloud	X	X	X	X	X	X
Collaboration btw Clouds	X	X	X	X	V	V

Chapter 3

SECURED WSN-INTEGRATED CLOUD COMPUTING

3 Secured WSN-integrated Cloud Computing

3.1 Overview

Our research scope falls into Wireless Sensor Network, Cloud Computing, and Security & Privacy for WSN ad Cloud, as shown in Figure 9. In this section, we present an overview of our proposed solution, Secured WSN-integrated Cloud Computing for u-Life Care, called SC^3.

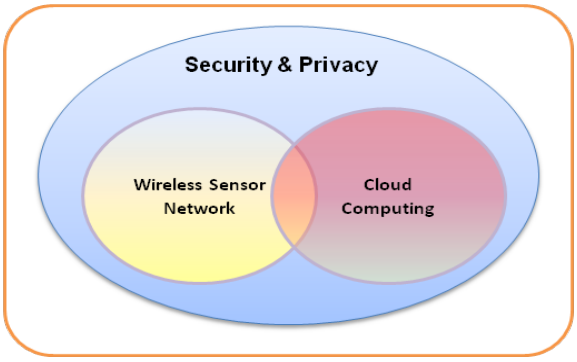


Figure 9 Our Research Scope

The abstract model is shown in Figure 10. We deploy a secure wireless sensor networks in u-Home environments for a purpose of monitoring and collecting sensor data. To enable u-Lice care applications, we propose an Activity Recognition engine module for u-Life care in WSN layer. This is very important engine to detect and report current user's activities for different purpose of life care services. The sensor data is transferred to Clouds by using sensor data dissemination and integration mechanisms. We provide a security and privacy control of data and applications stored in Clouds. Different Clouds can collaborate with each other by using our dynamic collaboration method. Numerous u-Life care services can access Clouds to provide better and low cost cares for end-users such as secure u-119 service, secure u-Hospital, secured u-Life care research, secure u-Clinic, etc. Figure 11 and Figure 12 shows the functional architecture and proposed architecture of the SC^3.

SC^3 is composed of the following modules: security for WSNs (trust management), security and privacy control for Clouds (authentication and access control), integration mechanism of wireless sensor networks to Clouds, sensor data dissemination mechanism, dynamic collaboration mechanism between different Cloud providers (CLPs), and activity recognition engine for u-Life care.

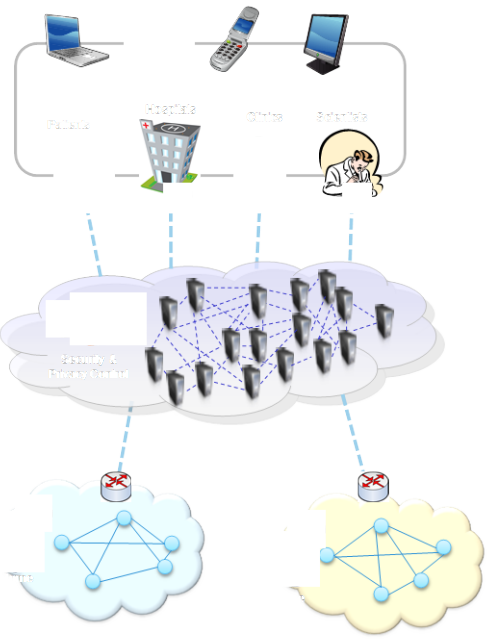


Figure 10 Overall Architecture of SC[3]

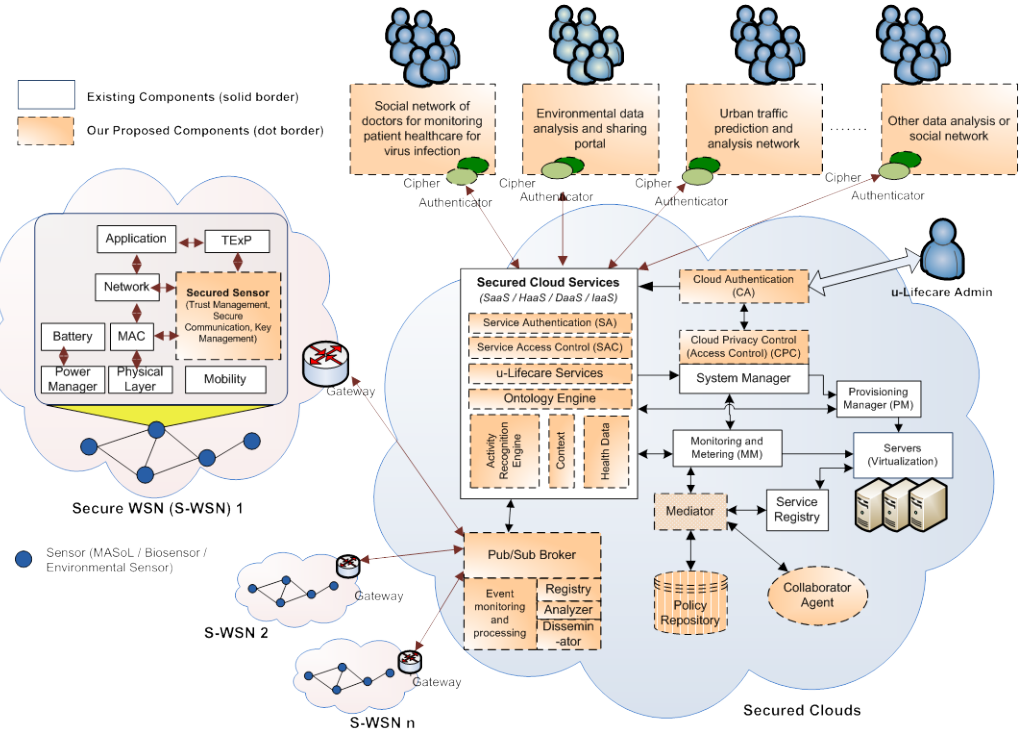


Figure 11 Functional Architecture of SC[3]

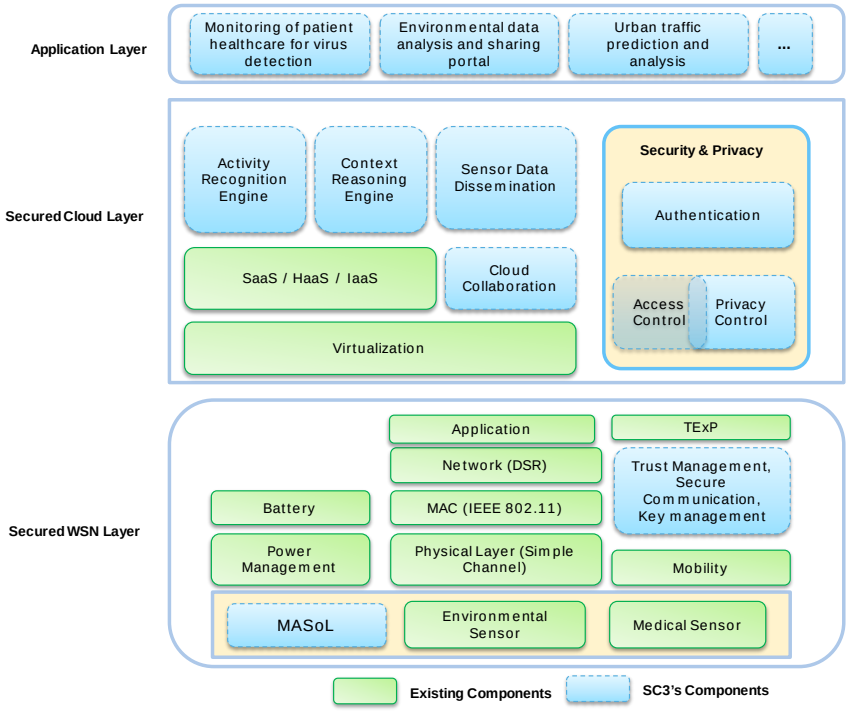


Figure 12 Layered Architecture of SC[3]

3.2 Challenges

Low resource sensors

Sensor nodes are very limited in term of energy, communication, and computation. Therefore, in order to make the algorithms feasible on sensor devices, they must be lightweight and energy-efficient.

A huge number of users, and it increases dramatically

As the number of users accessing Clouds increase dramatically, how to support individual users to declare their privacy preferences accurately.

Authentication method must be usable on various devices with wired or wireless-enable connection over the Internet.

Besides, appropriate privacy policy implementation is very hard. User must agree to provide his/her sensitive information which is not always possible

Data dissemination challenges

In case of dissemination of information to mobile clients, the mobility can cause their

access brokers to be changed, which can bring problems in dissemination of subscriptions and distribution of matching results.

Dynamic collaboration challenges

Finding appropriate group strategy to minimize collaboration cost in dynamic collaboration is really a major challenge.

3.3 Desired Components of SC^3

In the following sections, we present SC^3 in details. As shown in Figure 12, we propose SC^3 with the following components:

- Security and Privacy Control

Security for WSN including Trust Management
Security and Privacy Control for Clouds including Authentication, Access Control, Privacy Control
Integration of WSNs to Clouds
Sensor Data Dissemination Mechanism
Cloud Dynamic Collaboration Mechanism
Activity Recognition Engine for u-Life care

Chapter 4

SECURITY FOR WSN

4 Security for WSN

4.1 Group-based Trust Management Scheme

4.1.1 Introduction

A WSN is an essential technology for any health-care or life-care systems. Since life-care systems carries sensitive and private data, therefore security must be enforced in robust and reliable manner. Current security solutions of WSNs [5]-[9] are not capable of providing corresponding access control based on judging the quality of a sensor nodes and their services. This can only be achieved by in-cooperation of trust management scheme. The in-cooperation of trust in a security solution also provides other benefits such as:

Trust solves the problem of providing reliable routing paths that does not contain any malicious, selfish or faulty node(s).
Trust makes the traditional security services more robust and reliable by ensuring that all the communicating nodes are trusted during authentication, authorization or key management phases.

4.1.2 Problems of Existing Approaches

To the best of our knowledge, very few comprehensive trust management schemes (e.g. RFSN [10], ATRM [11] and PLUS [12]) have been proposed for sensor networks. Although, there are some other works available in the literature e.g. [13]-[16] etc., that discuss trust but not in much detail. Within such comprehensive works, only ATRM [7] scheme is specifically developed for the clustered WSNs. However, this and other schemes, suffer from various limitations such as these schemes do not meet the resource constraint requirements of the WSNs; and more specifically, for the large-scale WSNs. Also, these schemes suffer from higher cost associated with trust evaluation especially of distant nodes. Furthermore, existing schemes have some other limitations such as dependence on specific routing scheme, like the PLUS scheme works on the top of the PLUS R routing scheme; dependence on specific platform, like the ATRM scheme requires an agent-based platform; and unrealistic assumptions, like the ATRM assumes that agents are resilient against any security threats, etc. Therefore, these works are not well suited for realistic WSN applications. Thus, a lightweight secure trust management scheme is needed to address these issues.

4.1.3 Proposed Solution

Our proposed Group-based Trust Management Scheme (GTMS) scheme calculates the trust value based on direct or indirect observations. Direct observations represent the number of successful and unsuccessful interactions and indirect observations represent the recommendations of trusted peers about a specific node. Figure 13 shows our Trust Management component in general sensor node architecture.

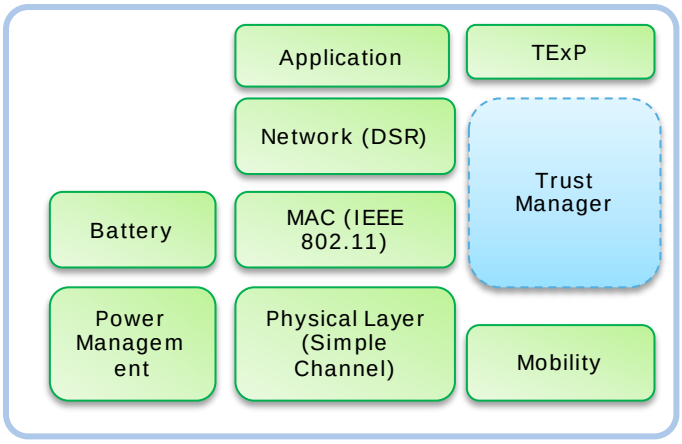


Figure 13 Sensor Node Architecture with our Trust Management Component

Interaction means cooperation of two nodes. For example, a sender will consider interaction as a successful interaction if he got assurance that the packet is successfully received by the neighbor node and he has forwarded it toward destination in an unaltered fashion.

First requirement of successful reception is achieved on reception of the link layer acknowledgment (ACK). IEEE 802.11 is a standard link layer protocol, which keeps packets in its cache until the sender received ACK. whenever receiver node successfully received the packet he will send back ACK to the sender. If sender node did not received ACK during timeout then sender will retransmit that packet.
Second requirement is achieved with the help of using enhanced passive acknowledgments (PACK) by overhearing the transmission of a next hop on the route, since they are within radio range [17].
If the sender node does not overhear the retransmission of the packet within a timeout from its neighboring node or overheard packet is found to be illegally fabricated (by comparing the payload that is attached to the packet) then the sender node will consider that interaction as an unsuccessful one. If the number of unsuccessful interactions increases, then the sender node decreases the trust value of that neighboring node and may consider it as a faulty or malicious node.

The proposed trust model works with two topologies. One is the intra-group

topology where distributed trust management is used. The other is inter-group topology where centralized trust management approach is employed. For the intra-group network, each sensor that is a member of the group, calculates individual trust values for all group members. Based on the trust values, a node assigns one of the three possible states: 1) trusted, 2) un-trusted or 3) un-certain to other member nodes. This three-state solution is chosen for mathematical simplicity and is found to provide appropriate granularity to cover the situation. After that, each node forwards the trust state of all the group member nodes to the CH. Then, centralized trust management takes over. Based on the trust states of all group members, a CH detects the malicious node(s) and forwards a report to the base station. On request, each CH also sends trust values of other CHs to the base station. Once this information reaches the base station, it assigns one of the three possible states to the whole group. On request, the base station will forward the current state of a specific group to the CHs.

Our group based trust model works in three phases: 1) Trust calculation at the node level, 2) Trust calculation at the cluster head level, and 3) Trust calculation at the base station level.

❖ ***Trust Calculation at the Node Level***

At the node level, a trust value is calculated using either time-based past interaction or peer recommendations. Whenever a node *y* wants to communicate with node *x*, it first checks whether *y* has any past experience of communication with *x* during a specific time interval or not. If yes, then node *x* makes a decision based on past interaction experience, and if not, then node x moves for the peer recommendation method.

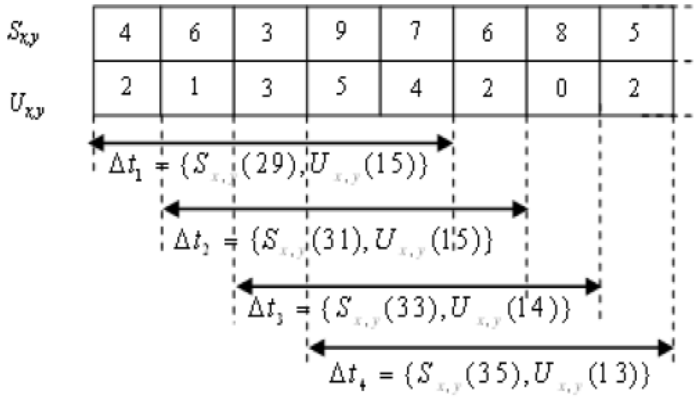


Figure 14 Sliding time window scheme of GTMS

Time-based Past Interactions Evaluation: Trust calculation at each node measures the confidence in node reliability. Here the network traffic conditions such as

congestion, delay etc., should not affect the trust attached to a node; this means that the trust calculation should not emphasize the timing information of each interaction too rigidly. Therefore, we introduce a sliding time window concept which takes relative time into consideration and reduces the effects of network conditions on overall trust calculation. If real-time communication is a requirement, as is the case in most real world applications, this timing window concept does not provide any hindrance when it comes to real-time delivery of packets. The communication protocol in such applications is always accompanied with time-stamps, and thus any node which delays the delivery of packets by taking advantage of the sliding timing window will be detected straightforwardly.

The timing window (Δt) is used to measure the number of successful and unsuccessful interactions. It consists of several time units. The interactions that occur in each time unit within the timing window are recorded. After a unit of time elapses, the window slides one time unit to the right, thereby dropping the interactions done during the first unit. Thus, as time progresses, the window forgets the experiences of one unit but adds the experiences of the newer time unit. The window length could be made shorter or longer based on network analysis scenarios. A sample scenario of the GTMS time window scheme is illustrated in Figure 14. The time window Δt consists of five units. During the first unit of Δt_1 , the number of successful and unsuccessful interactions is 4 and 2 respectively, and during the whole Δt_1 interval, the number of successful and unsuccessful interactions is 29 and 15 respectively. After the passage of 1st unit, the new time interval Δt_2, drops the interaction values which took place during the very first unit of Δt_1 (S = 4,U = 2) and only consider the values of last 4 units of Δt_1 plus values of one recent unit added on the right (S = 6,U = 2). With this time window information, the time-based past interaction trust value ($T_{x,y}$) of node *y* at node *x* that lies between 0 and 100, is defined as;

$$T_{x,y} = \left[100 \left(\frac{S_{x,y}}{S_{x,y} + U_{x,y}} \right) \left(1 - \frac{1}{S_{x,y} + 1} \right) \right] \quad (1)$$

where [.] is the nearest integer function, $S_{x,y}$ is the total number of successful interactions of node *x* with *y* during time Δt, $U_{x,y}$ is the total number of unsuccessful interactions of node *x* with *y* during time Δt. The expression $\left(1 - \frac{1}{S_{x,y} + 1} \right)$ in the above approaches 1 rapidly with an increase in the number of successful interactions. We choose this function instead of a linear function since such a function would approach very slowly to 1 with the increase in successful interactions; hence it would take considerably longer time for a node to increase its trust value for another node. In order to balance this increase in the trust value with the increasing number of unsuccessful interactions, we multiply the expression with the factor $\left(\frac{S_{x,y}}{S_{x,y} + U_{x,y}} \right)$ which indicates the percentage of successful interactions among the total interactions. Thus, this equation has an inbuilt capability of diminishing the effects of

a few wrong declarations of interactions that may be caused by any network traffic problems.

Figure 15 shows the behavior of time-based past interactions trust values against successful and unsuccessful interactions. When we do not get even a single successful interaction, the trust value remains 0. With an increase in successful interactions, the trust value increases, but stays humble if the number of unsuccessful interactions is also considerably high. For example, with 60 unsuccessful and 50 successful interactions, the trust value is 45.

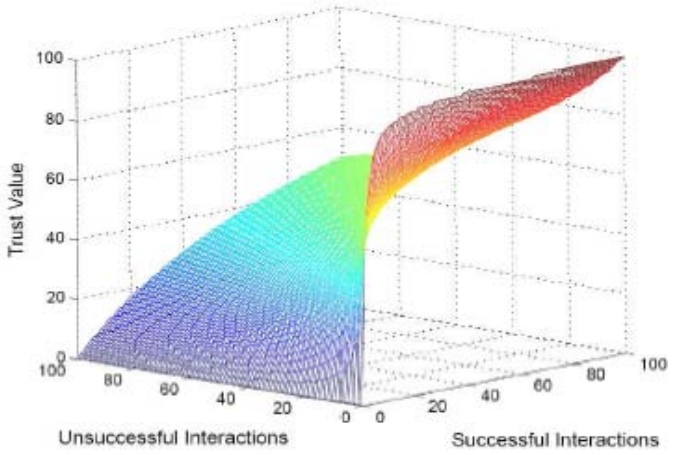


Figure 15 Time-based past interactions evaluation

After calculating trust value, a node will quantize trust into three states as follows:

$$M_p\left(T_{x,y}\right)=\begin{Bmatrix}\text{trusted} & 100-f\le T_{x,y}\le 100\\ \text{uncertain} & 50-g\le T_{x,y}<100-f\\ \text{untrusted} & 0\le T_{x,y}<50-g\end{Bmatrix} \quad (2)$$

where, *f* represents half of the average values of all trusted nodes and *g* represents one-third of the average values of all untrusted nodes. Both *f* and *g* are calculated as follows:

$$f_{j+1}=\begin{cases}\left[\frac{1}{2}\left(\frac{\sum_{i\in R_x}T_{x,i}}{|R_x|}\right)\right] & 0<|R_x|\le n-1\\ f_j & |R_x|=0\end{cases} \quad (3)$$

$$g_{j+1} = \begin{cases} \left[\frac{1}{5} \left(\frac{\sum_{i \in M_x} T_{x,i}}{|M_x|} \right) \right] & 0 < |M_x| \le n-1 \\ g_j & |M_x| = 0 \end{cases} \quad (4)$$

where [.] is the nearest integer function, R_x represents the set of trustful nodes for node *x*, M_x the set of un-trustful nodes for node *x*, and n is the total number of nodes that contains trustful, un-trustful and uncertain nodes. At startup, the trust values of all nodes are 50 which is an uncertain state. Initially *f* and *g* are equal to 25 and 17 respectively, although other values could also be used by keeping the following constraint intact: $f_i - g_i \geq 1$, which is necessary for keeping the uncertain zone between a trusted and un-trusted zone. The values of f and g are adaptive. During the steady-state operation, these values can change with every passing unit of time which creates dynamic trust boundaries as shown in Figure 16. At any stage, when $|R_x|$ or $|M_x|$ becomes zero then the value of f_j+1 or g_j+1 remains the same as the previous values (f_j and g_j). The nodes whose values are above 100 − *f* will be declared as trustful nodes (Eq. 2), and nodes whose values are lower than 50−*g* will be consider as untrusted nodes (Eq. 2). After each passage of time, Δ*t*, nodes will recalculate the values of *f* and *g*. This trust calculation procedure will continue in this fashion.

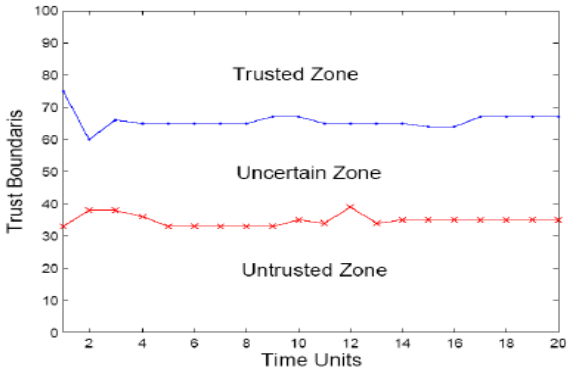


Figure 16 Adaptive trust boundaries creation

Peer Recommendations Evaluation: Let a group be composed of *n* uniquely identified nodes. Furthermore, each node maintains a trust value for all other nodes. Whenever a node requires peer recommendation it will send a request to all member nodes except for the un-trusted ones. Let us assume that *j* nodes are trusted or uncertain in a group. Then node *x* calculates the trust value of node *y* as follows:

$$T_{x,y} = \left[\frac{\sum_{i \in R_x \cup C_x} T_{x,i} * T_{i,y}}{100 * j} \right]; \; j = |R_x \cup C_X| \le n-2 \quad (5)$$

where, [.] is the nearest integer function, $T_{x,i}$ is the trust value of the recommender, and $T_{i,y}$ is the trust value of node *y* sent by node *i*. Here, $T_{x,i}$ is acting as a weighted value of the recommender that is multiplied with the trust value $T_{i,y}$, sent by recommender, such that the trust value of node *y* should not increase beyond the trust value between node *x* and the recommender node *i*.

❖ ***Trust Calculation at the Cluster-Head Level***

Here we assume that the CH is the SN that has higher computational power and memory as compared to other SNs.

Trust State calculation of Own Group: In order to calculate the global trust value of nodes in a group, CH asks the nodes for their trust states of other members in the group. We use the trust states instead of the exact trust values due to two reasons. First, the communication overhead would be less as only a simple state is to be forwarded to the CH. Secondly, the trust boundaries of an individual node vary from other nodes. A particular trust value might be in a trusted zone for one node whereas it may only correspond to the uncertain zone for another node. Hence the calculation of the global trust state of nodes in a group would be more feasible and efficient if we only calculate it using the trust states.

Let us suppose there are *n*+1 nodes in the group including the CH. The CH will periodically broadcast the request packet within the group. In response, all group member nodes forward their trust states, *s*, of other member nodes to the CH. The variable, s, can take three possible states: trusted, un-certain and un-trusted. The CH will maintain these trust states in a matrix form, as shown below:

$$TM_{ch} = \begin{bmatrix} s_{ch,1} & s_{1,ch} & \cdots & s_{n,1} \\ s_{ch,2} & s_{1,2} & \cdots & s_{n,2} \\ \vdots & \vdots & \vdots & \vdots \\ s_{ch,n} & s_{1,n} & \cdots & s_{n,n-1} \end{bmatrix}$$

where, TM_{ch} represents the trust state matrix of cluster-head *ch* and $s_{ch,1}$ represents the state of node 1 at cluster-head *ch*. The CH assigns a global trust state to a node based on the relative difference in trust states for that node. We emulate this relative difference through a standard normal distribution. Therefore, the CH will define a random variable *X* such that:

$$X(s_{i,j}) = \begin{cases} 2 & \text{when} \quad s_{i,j} = \text{trusted} \\ 1 & \text{when} \quad s_{i,j} = \text{un-certain} \\ 0 & \text{when} \quad s_{i,j} = \text{un-trusted} \end{cases} \qquad (6)$$

Assuming this to be a uniform random variable, we define the sum of *m* such random variables as S_m. The behavior of S_m will be that of a normal variable due to the

central-limit theorem. The expected value of this random variable is *m* and the standard deviation is $\sqrt{m/3}$. The CH defines the following standard normal random variable for a node *j*:

$$Z_j = \frac{\sqrt{3}\left(X(s_{ch,j}) + \sum_{i=1, i\neq j}^{m} X(s_{i,j}) - m \right)}{\sqrt{m}} \tag{7}$$

If $Z_j \in [-1, 1]$ then the node *j* is termed as un-certain, else if $Z_j > 1$, it is called trusted. If $Z_j < -1$, it is labeled as un-trusted.

Trust Calculation of Other Groups: During group-to-group communication, the CH maintains the record of past interactions of another group in the same manner as individual nodes keep record of other nodes. Trust value of a group is calculated on the basis of either past interaction or information passed on by the base station. Here we are not considering peer recommendations from other groups in order to save communication cost. Let us suppose CH i wants to calculate the trust value ($T_{i,j}$) of another cluster j. Then it can be calculated by using either time-based past interaction ($PI_{i,j}$) evaluation or by getting recommendation from the base station ($BR_{i,j}$) as shown below.

$$T_{i,j} = \begin{cases} \left[\dfrac{100(S_{i,j})^2}{(S_{i,j} - U_{i,j})(S_{i,j} + 1)} \right] & \text{if } PI_{i,j} \neq \varphi \\ BR_{i,j} & \text{if } PI_{i,j} = \varphi \end{cases} \tag{8}$$

If the cluster head does not have any record of past interactions within the time window, i.e., $PI_{i,j} = \varphi$, it requests the base station for the trust value.

❖ ***Trust Calculation at Base Station Level***

The base station (BS) also maintains the record of past interactions with CHs in the same manner as individual nodes do, as shown below.

$$T_{BS,ch_i} = \left[\frac{100(S_{BS,ch})^2}{(S_{BS,ch_i} - U_{BS,ch})(S_{BS,ch} + 1)} \right] \tag{9}$$

where, [.] is the nearest integer function, $S_{BS,ch}$ is the total number of successful interactions of BS with CH during time Δt, $U_{BS,ch}$ is the total number of unsuccessful interactions of BS with CH during time Δt.

Let us suppose there are |G| groups in the network. BS periodically multicasts request packets to the CHs. On request, the CHs forward their trust vectors, related to the recommendations of other groups based upon past interactions, to BS as shown below:

$$\overrightarrow{T_{ch}} = \left(T_{ch,1}, T_{ch,2}, \cdots, T_{ch,|G|-1} \right)$$

On reception of trust vectors from all the CHs, the base station will calculate the trust value of each group in a manner shown below:

$$T_{BS,G_1} = \left[\frac{\sum_{i=1}^{|G|-1} \left(T_{BS,ch_i}\right)\left(T_{G_i,G_1}\right)}{|G|-1} \right], \cdots, T_{BS,Gm} = \left[\frac{\sum_{i=1}^{|G|-1} \left(T_{BS,ch_i}\right)\left(T_{G_i,G_{|G|}}\right)}{|G|-1} \right] \quad (10)$$

where, $T_{BS,ch}$ is the trust value of the CH *i* at the base station, $T_{Gi,G1}$ is the trust value of group G_1 at group G_i and $|G|$ represents the total number of groups in the network.

Implementation Interface Description

Proposed GTMS module has four external interfaces as shown in Figure 17.

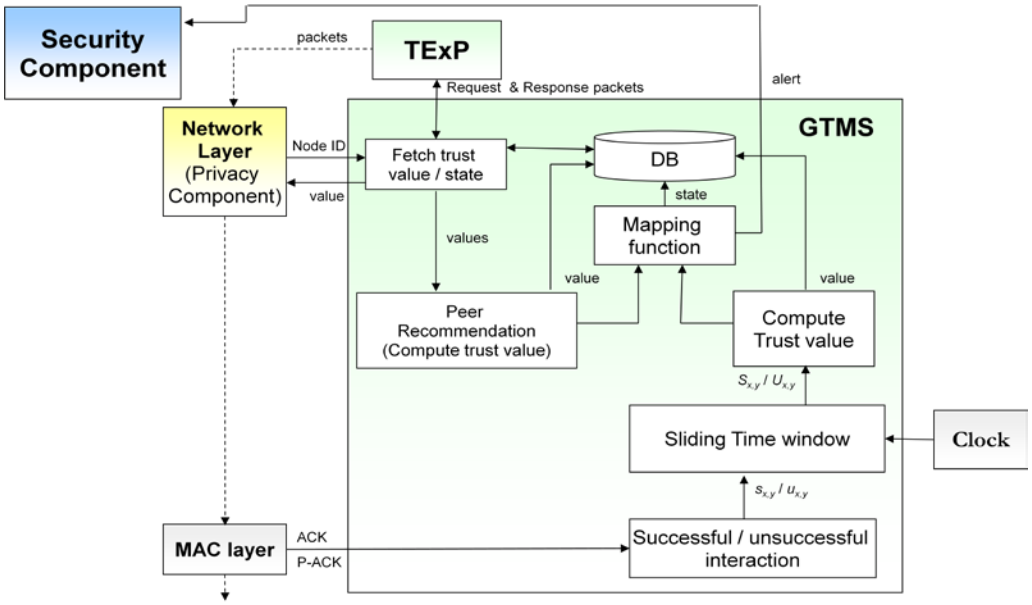


Figure 17 Interfaces of trust component

MAC interface: From the MAC layer, GTMS component receives link layer acknowledgment (ACK) and enhanced passive acknowledgment (P-ACK) for transfer of each packet. Based on these two information, the GTMS module considers an interaction as a successful or an unsuccessful one. This information will be further recorded in the sliding time window. With this time window information, the time-based past interaction trust value of the other node is calculated.

Network interface: Whenever a routing protocol (e.g IRL or r-IRL) needs to select trusted next hop node for the purpose of forwarding packets, it first interacts with the GTMS module. During the initialization phase, IRL and r-IRL protocols provide node identities to the GTMS module. GTMS module tells IRL and r-IRL protocols that

which neighboring nodes are trusted. Based on this information, the routing protocol makes routing decisions.

Exchange interface: Whenever GTMS module needs recommendations from other nodes; it sends request packets via generic Trust Exchange Protocol (TExP). Based on the recommendation received via TExP protocol, it computes trust value.
Alert interface: Whenever GTMS module detects some malicious node, it will send alert message to the security component.

Chapter 5

SECURITY AND PRIVACY CONTROL FOR CLOUD

5 Security and Privacy Control for Cloud

5.1 Image Feature-based Authentication Module

5.1.1 Introduction

Most of current Web-based applications are using password-based authentication mechanism. When we use passwords we assume that:

Our chosen passwords are hard to guess
We are using a malicious software free terminal
No one is shoulder surfing

Unfortunately this is not true all of the time. We therefore need to find an authentication mechanism that is immune to the above three forms of attack. It is easy to see that the other alternatives such as PIN numbers, Biometrics, etc do not do much better. The protocols which insure secure authentication when the above mentioned conditions are not true are called as Human Identification Protocols.

5.1.2 Problems of Existing Works

The first work on human identification dates back to [5]. Since then a lot of other schemes have been proposed in literature [19]-[24]. Some of them were broken in [19], [25]. While most of them involve some numerical calculations like [5] and the HB protocol [22] they can be implemented using some graphical interface employing pictures as memory aids. We can categorized the human identification protocols into two broad categories: Protocols built to be secure against general eavesdropping adversaries and protocols secure against only “guessing adversaries” i.e. Adversaries who do not see the user's input and hence try to guess the secret or impersonate the user without any prior knowledge. Protocols mentioned so far fall in the first category. They have a drawback, however, that they involve extra computation from the user. As an example, in the HB protocol [28], the user is required to compute bit-wise binary multiplication for some number of bits in every iteration. This may not seem much but to obtain a higher level of security, the number of computations increase significantly.

In the second category, the most well known example is the traditional password based authentication system. Others include purely graphical schemes like déjà vu [26], Passface [27], Point & Click [28] and [29] that require little or no numerical

computation whatsoever. The basic theme of [26] and [27] is to present the user a series of pictures, a subset of which are the secret pictures. The user is authenticated if its selection of the secret pictures among the given set of pictures is correct. On the other hand, in [28] the user is authenticated if it clicks on the correct secret location in the given picture. [22] works similarly by letting the user draw the secret symbol or figure on a display device. Evidently, these purely graphical schemes are not secure against ``peeping'' attacks [23]. Anyone observing the actions of the user can find out the secret in no time. For a detailed account of all the schemes, see [23].

5.1.3 Proposed Solution

The main theme of our idea is the famous expression: A picture describes a thousand things. A picture illustrates a huge number of attributes (features). The picture in Figure has for example the following main things: Hat, Lena, female, model, human, nose, eyes, hair, purple, mirror, image processing, etc. And the list goes on. Based on this assumption our main protocol structure is as follows:

User is given a picture and a secret “attribute”
Answer “yes” if the picture contains the attribute, otherwise “no”

The task of the adversary is to guess what attribute is being answered by the user. We make things harder for the adversary by “Shuffling” the answers to the pictures based on a secret string of picture positions which is also shared as a secret between the user and the server. This string need not to be more than 3 or 4 digits long.

The protocol can now be described by the following steps:

User and server share a secret feature and 3-digit secret string (e.g. 341)

Repeat 3-times:
Server sends 10 pictures to the user
User sends response string (e.g. 101) corresponding to the pictures in the secret string
If all bits are correct then **accept**
Else
if 8 out of 9 bits are correct then repeat the above procedure once (for possible system mistake) and **accept** only if all 3 bits are then correct
Else reject

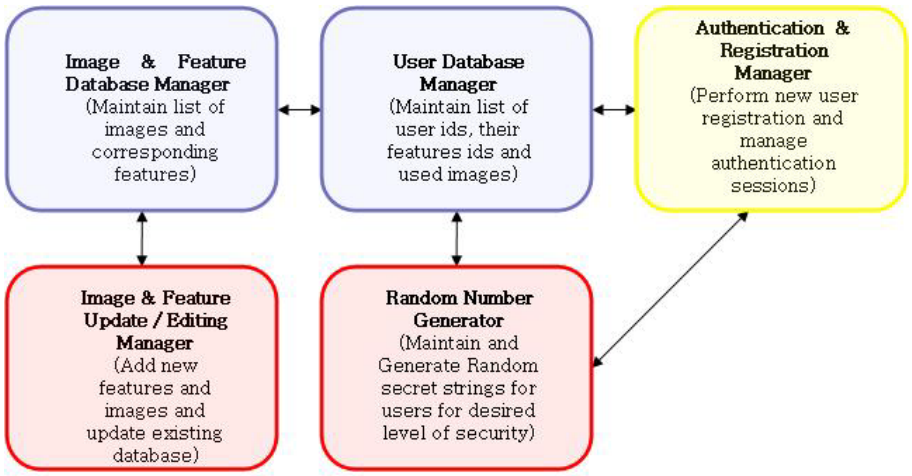


Figure 18 Functional diagram of the authentication module.

We can display 10 pictures at a time. With probability ½ each picture satisfies the question. The legitimate user has to answer '3' of them based on the secret string e.g. 341. Protocol can be run thrice to get a security of $2^{-9.}$ This is almost as secure as a "truly" random 4-digit pin. Every picture is shown only once to avoid the adversary gaining knowledge.

In the case of mobile Terminals, we know that they possess smaller screens. Thus they cannot display many pictures at the same time. But they have an advantage too: They are not easily susceptible to shoulder surfing. Thus we need not show the extra images to the user. These images can just serve as decoy images when sent from the server to the user and will be discarded at the user's terminal when the user inputs its secret string at the time of authentication in the mobile device. This string is not stored in the device in case the device is stolen. Thus our protocol extends beautifully to the mobile terminals.

The functional diagram is shown in Figure 18.

A brief description of the modules is given below:

Image and Feature Database Manager: The image and feature database manager maintains a database of images and an index file comprised of feature IDs and image ID's corresponding to the Feature IDs.

User Database Manager: This manager maintains a list of user IDs, user's features and their ids, user's secret strings and a list of used images for the user's authentication sessions.

Authentication Manager: This is the authentication and registration manager which registers a new user by providing it with a set of selectable features. It also

retrieves the feature id of an existing user and revokes an old feature/string pair.

Random Number Generator Manager: The random number generator manager uses the Cryptographic Library in our system to manage the security parameters such as the length of the secret string for a particular user. It also helps to generate pseudorandom strings for a new user based on a pseudorandom number generator (PRNG) present in the Cryptographic Library.

Image and Feature Updater: This sub-module adds new features to the image database. It does this by selecting a word and finding all its possible synonyms or terms with similar meanings. It then retrieves images from the web satisfying the features. Finally it updates the image database by adding new and/or editing existing images. Human administration is also required to check the validity of new features and corresponding images.

5.2 Activity-Oriented Access Control

5.2.1 Introduction

u-Life care domain demands a big security challenge to researchers. Protecting the confidentiality of Cloud data, while at the same time allowing authorized users to access conveniently, is a core issue in u-Life care. The need of delivering information such as patient medical records at the point-of-care is a primary factor in managing the healthcare system efficiently. Current healthcare systems require considerable coordination efforts of physicians to locate relevant documents according to their specific activity. Physicians must have access to a large amount of proper data in different locations that's often tied to their specific work. Users must constantly log in and out of devices, start and stop sets of applications, look for what types of information needed for their care and browse information repeatedly. For example, at the patient's bedside, the physician enters notes in the electronic patient records. During the radiology conference, a physician studies X-ray images with a radiologist. In the morning conference, the physician discusses proper medication with colleges while browsing medicine catalogs. Later, the lab releases a blood sample result, and the physician must study it with colleagues. A proper access control model to increase efficiency of professional hospital work is essential.

5.2.2 Problems of Existing Work

Current solutions to this problem (mostly built on static RBAC models [30]-[36]) are application-dependent and do not address the intricate security requirements of healthcare applications. The healthcare industry requires a flexible, extensible context-aware access control, and dynamic authorization enforcement. Most access control models exploit user identity/role information to determine access

permissions [30]-[32]. The policy specification of these models tightly couples identity/role of users with their permissions. This coupling does not support user work. Other works use context as a foundation to authorize access [34][35]. However, their concept of context is general, for example location context, time context, system context, etc. It does not precisely specify user's activities. In order to provide information and services to users at the point-of-care, they must spend a considerable effort to modify and adapt those approaches into reality.

In this work, we explore the use of user activity concept to go one step further. We propose an Activity-Oriented Access Control (AOAC) scheme to support user activities in ubiquitous hospital environments. In AOAC, access to computer system objects is based on a user's activity. A user is allowed to carry out an activity if s/he holds a number of satisfactory attributes such as his/her roles, assignments, etc under a certain environment circumstances. For example, doctor Bob is assigned to treat patient Carol; to carry this activity, Bob needs access permissions to hospital information such as Carol's medical record, x-ray image, blood test result.

5.2.3 Proposed Solution

Based on characteristics of organizational authorization policy, we abstract the access control model in three levels: user level, activity level, and privilege level, as illustrated in Figure 19. The nature of organizational authorization is who is allowed to do what, for example doctor Bob is permitted to carry out patient treatment of pneumonia. Therefore, by associating users with activity, we can easily match the authorization model into real environments. On the other hand, in order to accomplish an action, a user needs access permissions to a number of resources. By connecting each activity with access permissions, the proposed model highly supports user activity.

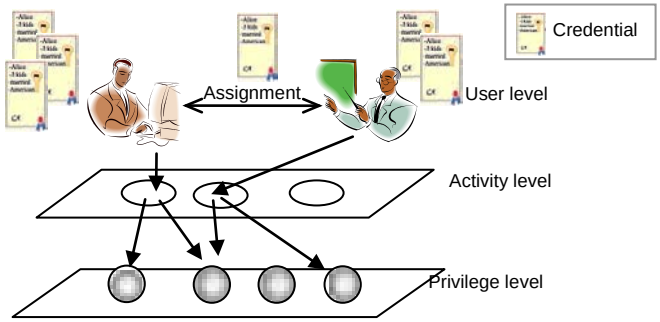


Figure 19 Abstract levels of AOAC

Each user holds a number of credentials [37] specifying his attributes such as hospital role, experience, assignment. A credential can be a certificate/qualification (e.g. doctor license), a hospital role (e.g. screening nurse), or an assignment from another user (e.g. a doctor is on leave, so he assigns to another doctor to diagnose his

patients during his absence). A user is authorized to perform a certain activity if the conditions are satisfied. We define the condition as *activity activation rule*. Each activity is associated with a number of access privileges. Those access privileges are needed to support user to accomplish his activity. For example, the activity '*prescribe medicine for patient Carol*' requires access permission to medical record, x-ray images, blood test results, and medicine charts. We define this rule as *permission activation rule*.

❖ ***Privilege Delegation via Digital Credentials***

Basically, *privilege delegation* is a term to indicate that user *A* delegates to user *B* a particular privilege. For example, privilege to diagnose a patient. In AOAC, we define this as *assignment*, where *A* is an *assigner* and *B* is an *assignee*. *Assignment* is a similar concept with *appointment* in [36]. It occurs when a user grants a digital credential that directly or indirectly allows some user to perform one or more activities. The credential content may be an assignment of activities (*direct delegation*), or may be an assignment of role, qualification, etc so that the user may use to activate some activity (*indirect delegation*). Our delegation approach differs from *appointment* in several aspects. Firstly, user *A* may not only delegate the object right to *B*, but *B* may also delegate the right to another user *C*, and so on. We call this *multi-step delegation*. Secondly, our privilege delegation may be *restricted* or *unrestricted*. It means that *A* may restrict how *B* can further delegate its access right. Delegation in *appointment* approach is only concerned with how to grant another user a credential to activate one or more roles without any concern about further delegation or restriction.

Digital credentials are meant to be the digital equivalent of paper based credentials. Digital credentials prove something about their owner (e.g. person position) or deliver some information of their owner (e.g. job assignment). We define a credential in a form of $CRED(USERS_1,USERS_2)$, where $USERS_1$ is a subset of users who can grant the credential, and $USERS_2$ is a subset of users who can be granted.

It is essential to restrict which users can grant credentials and which users can be granted. This contributes two advantages. Firstly, for hospital policies, this is important because, for instance, some sensitive credentials must be only granted by superiors and only be granted to seniors in the hospital. For example, only oncologists are allowed to give treatment for cancer patients, others those do not have specialty in cancer should not be permitted to get this assignment. Secondly, this avoids mistakes or abuses of legitimate users and prevents unauthorized assignments. For example, if a user is not *screening nurse,* she cannot assign patients to doctors for treatment; or if a user is not a doctor, he cannot designate a nurse to prescribe medicine.

We define an *assigner* is defined as a user who initiates a process of an assignment and grants s credential to another user. In order to initiate a process of an assignment,

the *assigner* must be indicated in the user set $USER_1$ ($assigner \in USER_1$) and the credential is able to be delegated; an *assignee* is defined as a user who is delegated a credential. It is essential that the *assignee* is specified in the user set credential's $USER_2$ ($assignee \in USER_2$). An *assignee* in a process of assignment may be an *assigner* in another process of assignment if the *assignee* is specified in the user set $USER_1$ ($assignee \in USER_1$) of the credential and transitive step must not be less than 1 ($N \geq 1$).

We define an *assignment* is a process that an *assigner A* sends a credential to an *assignee B.* Formally:

$$A \rightarrow B: \text{assignment (CRED (USERS}_1, \text{USERS}_2), N)$$

where *N* is number of further assignment that an assigner can delegate to an assignee (*N=1,2,3,…*); if *N=1, then B* cannot grant this credential to anyone; if $N=\infty$: anyone possessing the credential *CRED* can grant it to another user (for sure, *B* must be indicated in *CRED's* $USERS_1$).

For an example, we assume that doctor John is on leave, so he grants permission to doctor Peter to diagnose his patients during his absence. The assignment is formally described as follows:

John→Peter: assignment (DIAGNOSE ({doctors, nurses}, {doctors, nurses}), 1).

❖ ***Privilege Revocation***

Privilege revocation is very important. There are many situations that credentials should be revoked. For example, Alice delegated a credential to Bob for a particular business; if the business is accomplished, or Bob is transferred to another department, the credential should be revoked immediately. There are different reasons and different ways to revoke delegated credentials. Privilege revocation can be done in four ways[36]: by its assigner only; by anyone active in the credential; by assignee's resignation; or by rule-based system revocation: revocations can be carried out by the system itself. There are different ways to revoke a privilege: time duration on the credential is expired; constraint on the credential is violated; the task is completed; revoked at the end of the assigner session or the assignee's session.

We propose two types of revocations: *single-step revocation* and *multi-step revocation*. Single-step revocations are applied for single-step delegations. It means that Alice only delegates a credential to Bob without any further delegation from Bob. Multi-step invocations are applied for multi-step delegations. However, single-step revocations could be considered as a case of the multi-step revocation with the number of delegation step is one. There are different ways for cascading revocations. However, one of the simplest ways is based on credential identifiers (ids) to revoke them. It means that the original assigner (Alice) attaches a unique id to each

credential. Whenever she wants to revoke, she just needs to indicate the credential's ids and the system will consider those credentials as invalid ones.

❖ ***Activity Activation Rules***

In order to perform an activity, the user must satisfy the conditions of the *activity activation rule*. We use Prolog-like expression to formulate our *activity activation rule* as follows:

$$ACT \vdash CON_1, CON_2, \ldots, CON_n$$

An example of *activity activation rules* is given as follows:

treating_patient (Bob, Carol) ├ med_doctor(Bob), screening_nurse(Alice), patient(Carol), treating_assignment(Alice,Bob)

i.e. Bob is allowed to treat *patient Carol* if he holds a medical doctor license, and is appointed by screening nurse Alice to treat Carol.

It is notice that CON_i could be an attribute that a user holds including privileges to perform an activity, privileges to access a resource, etc. It is not restricted to only user's properties or characteristics.

❖ ***Permission Activation Rules***

Whenever a user activates an activity, corresponding permissions are automatically activated if a number of context constraints are satisfied. We define context constraints as: *Context constraints* are defined as any requirement about contextual information such as time, location, etc. Context constraints play a key role to specify context-sensitive policies. In organizations, this is very important to restrict user's access and invoke user's privileges if contextual requirements are not met.

Permission activation rules are formulated as follows:

$$PERM \vdash ACT, CC_1, CC_2, \ldots, CC_m$$

i.e. if user can activate an activity *ACT* under satisfied context constraints CC_1, $CC_2,\ldots,CC_m$, then s/he is granted the corresponding permissions *PERM.*

For example, if Alice is allowed to carry out activity "*taking_note*", then she is permitted to access EPR of Carol during working time.

read_EPR (Alice , Carol) ├ taking_note(Alice), time(11:00)

User's attributes, activities and activity activation rules are stored on a Lightweight

Directory Access Protocol (LDAP) server. Permission activation rules are defined by eXtensible Access Control Markup Language (XACML) standard. The system design is depicted in Figure 20. To be compliant with XACML standard, AOAC includes a Policy Enforcement Point (PEP), and a Policy Decision Point (PDP). PEP performs access control by making decision of requests and enforcing authorization decisions. PDP evaluates applicable policy and renders and authorization decision. The PDP is composed of three sub-components: User-Attribute Manager (UAM), which retrieves user's attributes according to user identifier (uid) from user's attributes LDAP server; Attribute-Activity Manager (AAM), which matches user's attributes to a set of allowed activities; and Activity-Permission Manager (APM), which retrieves all access privileges for given activities from XACML policies. The Admin Tool is used by the system administrator to define activities and policies. In AOAC, activities are clearly defined for each service through the Admin Tool.

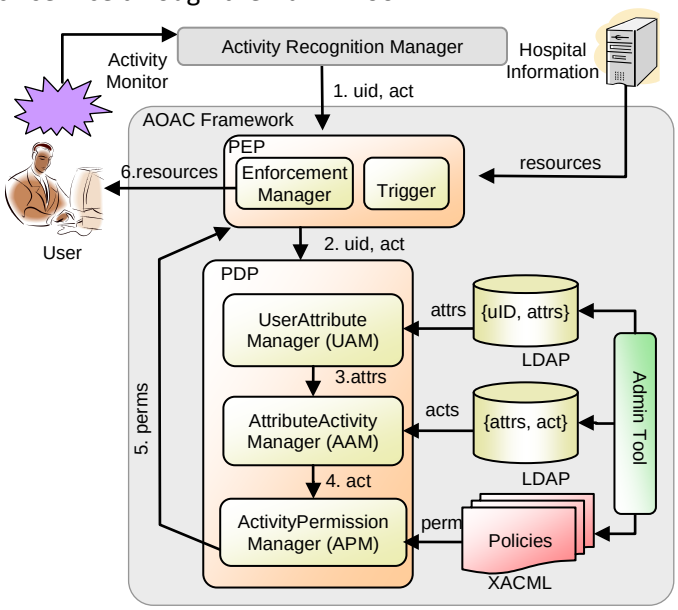


Figure 20 AOAC System Design

All possible activities in the hospital are predicted and predefined in advance. If there is a new activity, it can be added through this tool. The system operates by the following steps:

1) ARM provides user's activity information to AOAC by gathering raw contextual data related to user activity, producing high level context, and then reasoning user activity.

2) PEP forwards user id (*uid*) and current activity (*act*) to UAM.

3) UAM queries all attributes (*attrs*) matched to *uid* from LDAP server. It then sends those attributes to AAM.

4) According to user's attributes, AAM looks up LDAP server and makes decision if the

user is allowed to perform the activity (*act*). If yes, AAM forwards the activity (*act*) to APM.

5) At APM, corresponding permissions (*perms*) for the activity are achieved by checking the system policies. PDP then sends list of permissions to PEP.

6) At PEP, the Trigger Module is invoked to retrieve data. Then, data is sent to the user's device.

5.3 Privacy Control

5.3.1 Introduction

In recent years, privacy data is in spotlight since the quantity of privacy data is growing at a tremendous speed and threats are exist ubiquitously by various ways. In Cloud, privacy data processed outside the enterprise brings with it an inherent level of risk. For example, user data for u-Life care system like personal information, context information (location, activity, etc), medical data (medical history, drug information, medical health record, etc) are highly sensitive, people do not want to disclosure it to the public, they want to control how to release information. So we must consider privacy carefully in Cloud-based u-Life applications.

5.3.2 Problems of Existing Approaches

Existing privacy control mechanisms [38]-[40] have some limitations. For example, in [38], authors use the concept of Discrete Box as the central unit of the middleware architecture. The Discreet Box provides the user with appropriate access rights in order to be able to update or delete his personal data and the associated metadata. Every piece of personal information on its way to the service providers is filtered by the Discreet Box, so that the danger of unauthorized access and misuse is eliminated. However, their work does not mention in which way the data stored, it only considering individuals' personal data rather than unified management. In [39], authors propose a family of Privacy-aware Role Based Access Control (P-RBAC) model that provides full support for expressing highly complex privacy-related policies, taking into account features like purposes and obligations. But P-RBAC is not designed to support individual user preferences and here the data is stored in a tree structure, which cannot illustrate the complicated relation among the data set.

5.3.3 Proposed Solution

Basic definitions used in Our System

Definition 1 (Role): Let R be the set of roles in our system, denoted by $R = (r_1, r_2 \ldots r_i)$ where r_i denotes any role.

Definition 2 (Privacy Data Graph): Let PV_j be the privacy data graph of r_i.as show n in Figure 21. A privacy data graph comprises a set *V* of vertices together with a set *A* of arcs (directed edges).

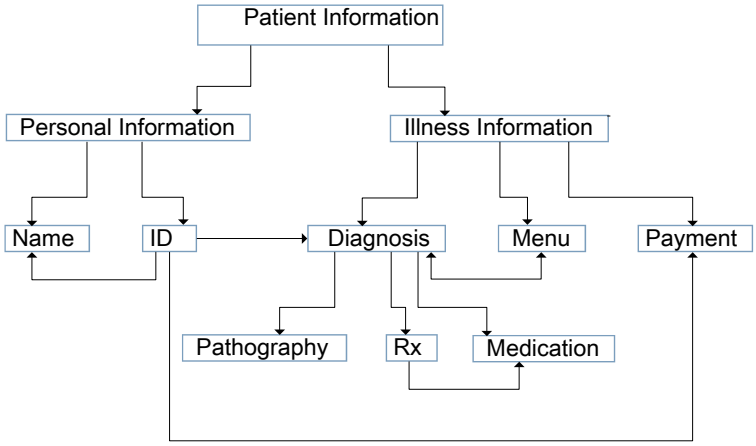


Figure 21 A directed graph Di of patient role ri

Definition 3 (Personal Privacy Preference Policy): For each user taking a specific role, we define a set to store personal privacy preferences for different roles. A $PV_i = (pv_1, pv_2 ..., pv_n)$ is a user's personal privacy preference against r_i. Each pv indicates a personal data element or an abstract personal data element

Definition 4 (Access Policy Graph): Given a pair of roles <r_i,r_j>, two directed graphs D_{ij} and D_{ji} are defined by the administrator of r_i and the administrator of r_j respectively. D_{ij} denotes r_i' privacy data access policy for r_j. D_{ji} denotes r_j's privacy data access policy for r_i.

Our proposed solution aims at utilizing a definition of graph data structure, which not only supports privacy-aware role based access control but also allow individual user preference. We believe that the definition of graph data structure solution to the problem of how to present the relationship among data items in privacy-aware systems may have a great potential. The architecture of our proposed system is presented in Figure 23. There are five major phases in our system as shown in Figure 23.

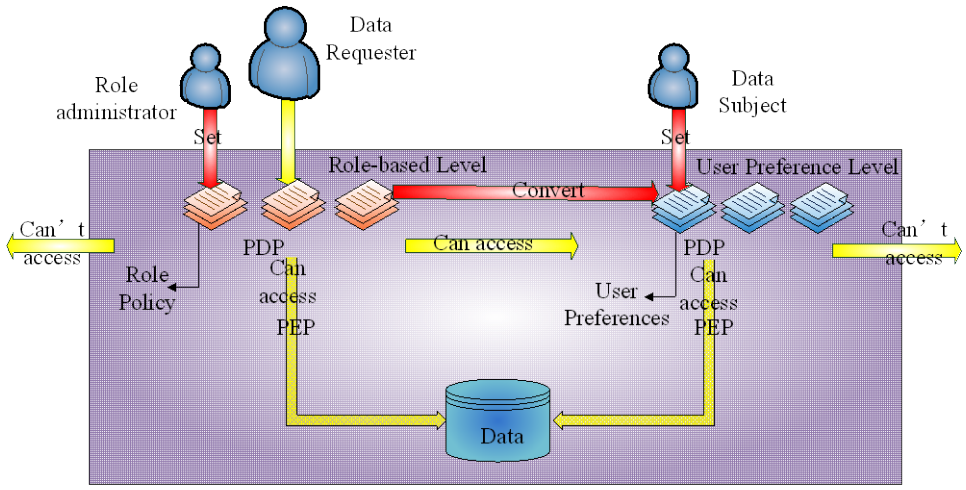


Figure 22 Our proposed Privacy control system

The first phase is the system design phase. It is also called privacy-aware system engineering phase. In this phase, the designer of system should achieve following goal: privacy requirement analysis, DBD (database design), role design, privacy language specification, security design and so on. After first phase, the role set R and D_i for corresponding r_i are distributed to the administrators of roles.

In second phase, based on their own situation, the administrators assign their privacy graphs. This is also a process specifying the rights and obligations for role members. In order to avoid unexpected data disclosure or conflicts in definition phase, we propose a detection algorithm. However, those privacy graphs set by administrators are not directly sent to individual users. Each D_{ij} is converted to default PV_j. Here we propose another algorithm for conversion which converts the rules of role to individual privacy preference set. Then those PVs are distributed to corresponding users according to their roles.

The third phase contains two parts: setting personal privacy preference, accomplished by personal data owner, and setting privacy access policy accomplished by the person who wants to access personal data. If a relationship between two users is going to be established, via PDP (policy decision point), system checks whether their personal preference and privacy policy is matched or not. This matching process is the fourth step. If matched, an agreement will be established and stored in the system.

The last step is checking enforcement via PEP (policy enforcement point) when personal data access happens. The foundation of PEP is the agreement made in previous step.

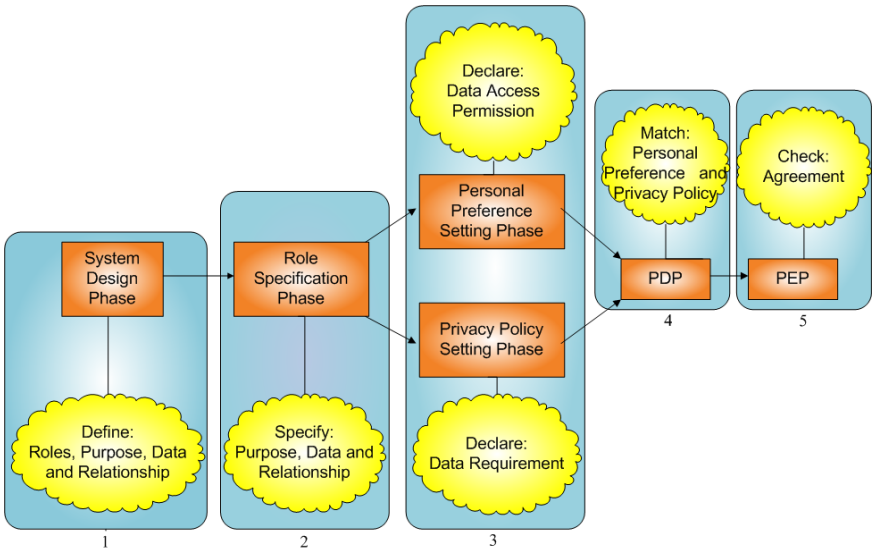


Figure 23 The Work flow diagram of proposed system

In conclusion, benefits of our approach include:

Properly storing sensitive data in a graph data structure can clearly illustrate the relation among the data items. As some vulnerability may be raised after certain sensitive data are disclosed, our data storage structure successfully solved this problem by reducing disclosure of data in a minimum way.

Apart from that, it also facilitates checking the relativity among the data items, avoiding conflicts and unconscious data disclosure in any phase.

Support Role Based Access Control (RBAC) that simplifies the specification and management of individual user or enterprise, especially in the case of large amounts of users.

Allow users to set their privacy preference by adding, deleting or modifying some data on the basis of the role storage structure. With such capabilities user can declare their privacy preferences more accurately and in a flexible way.

Chapter 6

WSN-CLOUD INTEGRATION

6 WSN-Cloud Integration

6.1 Introduction

In the past few years, wireless sensor networks (WSNs) have been gaining increasing attention to create decision making capabilities and alert mechanisms, in many Life care application areas including Life care monitoring for patients, environmental monitoring, pollution control, disaster recovery, military surveillance etc.

Collection, analysis (knowledge processing, ontology reasoning etc.), storing and disseminating of these sensor data is a great challenge since sensor nodes constituting a WSN have limited sensing capability, processing power, and communication bandwidth. However, there is a lack of uniform operations and standard representation for sensor data.

6.2 Problems of Existing Works

Currently there is no framework to support the integration of WSNs to Cloud. There are many challenges exist to enable this framework as the entire network is very dynamic.
On the WSN side, sensor or actuator (SA) devices may change their network addresses at any time
Wireless links and SA devices are quite likely to fail at any time, and rather than being repaired, it is expected that they will be replaced by new ones.
Besides, different Cloud applications can be hosted and run on any machines anywhere on the cloud. In such situations, the conventional approach of using *network address* as communication means between the SA devices and the applications may be very problematic because of their dynamic and temporal nature.
Moreover, several Cloud applications may have an interest in the same sensor data but for different purposes. In this case, the SA nodes would need to manage and maintain communication means with multiple applications in parallel. This might exceed the limited capabilities of the simple and low-cost SA devices.

6.3 Proposed Solution

We propose a content-based publish/subscribe (pub/sub) [1] broker model on the Cloud that integrates WSNs to Cloud efficiently and effectively. The framework is shown in Figure 24. The terminologies used to describe the system architecture are listed in Table 2. In this framework, sensor data or events are delivered to the

consumers or applications on the Cloud *not based on their network addresses*, but rather as a *function of their contents and interests.* The pub/sub broker is located in the Cloud to gain high performance in terms of bandwidth and capabilities.

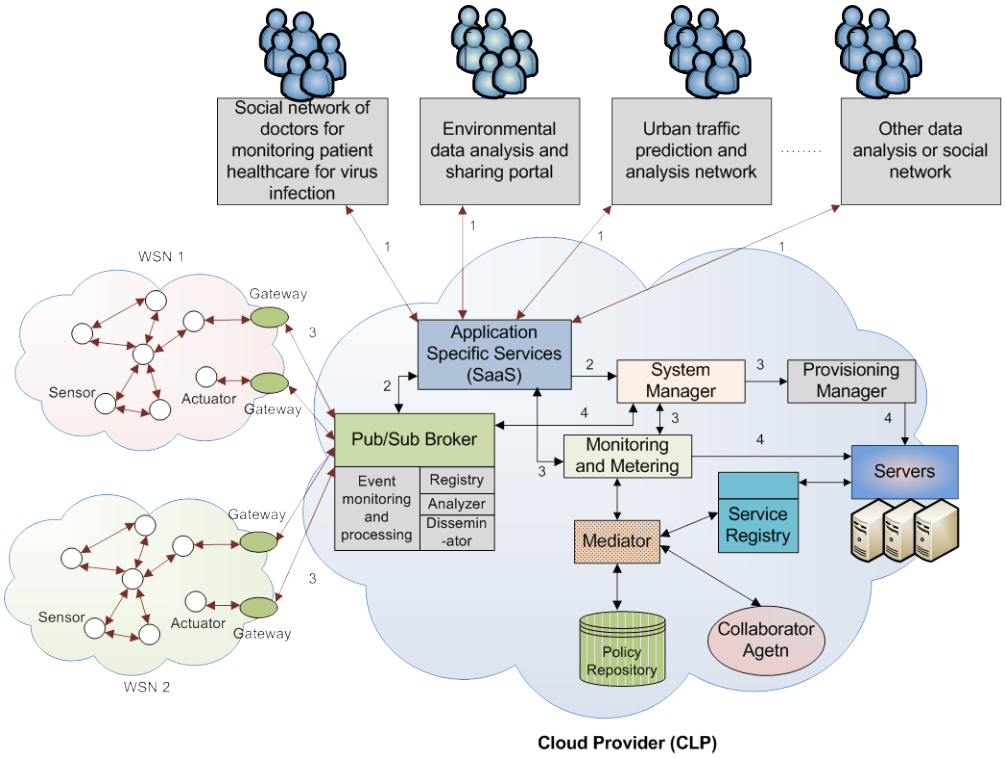


Figure 24 A Framework of WSN – Cloud Integration

Table 2: List of commonly used terms

Terminology	Description
Gateway	Sensor data comes to the pub/sub broker through gateway
Pub/Sub Broker	Responsible to monitor, process and deliver events to users (registered) through SaaS applications
SaaS (Software as a Service) Application	SaaS application which combines data and application together and runs on the Cloud server
System Manager (SM)	Manages the computer resources
Provisioning Manager (PM)	Carve out the systems from the cloud to deliver on the requested service. It may also deploy the required images.
Monitoring and Metering (MM)	Tracks the usage of the primary cloud resources as well as the resources of collaborator CLP's so the resources used can be attributed to a certain user.

Service Registry (SR)	Discovers and stores resource and policy information to local domain
Mediator	Responsible for negotiation among the collaborating CLPs and management of operations within a VO
Policy Repository (PR)	A storage of Web server, mediator and VO policies
Collaborator Agent (CA)	A resource discovery module in the collaborating CLPs environment

The pub/sub broker has four components describes as follows:

Sensor Stream or Event monitoring and processing component (SMPC): The sensor stream comes in many different forms. In some cases it is raw data that must be captured, filtered and analyzed on the fly and in other cases it is stored or cached. The style of computation required depends on the nature of the streams. So the SMPC component running on the cloud monitors the event streams and invokes correct analysis method. Depending on the data rates and the amount of processing that is required, SMP manages parallel execution framework on Cloud.
Registry Component (RC): Different SaaS applications register to pub-sub broker for various sensor data required by the community user. For each application, registry component stores user subscriptions of that application and sensor data types (temperature, light, pressure etc.) the application is interested in. Also it sends all user subscriptions along with application id to the disseminator component for event delivery.
Analyzer Component (AC): When sensor data or events come to the pub-sub broker, analyzer component determines which applications they are belongs to and whether they need periodic or emergency deliver. The events are then passed to the disseminator component to deliver to appropriate users through SasS applications.
Disseminator Component (DC): For each SaaS application, it disseminates sensor data or events to subscribed users using an event matching algorithm (described in section 6). It can utilize Cloud's parallel execution framework for fast event delivery.
The pub-sub components workflow shown in Figure 25 is described as follows:

① Users register their information and subscriptions to various SaaS applications which then transfer all this information to pub/sub broker registry.
② When sensor data reaches to the system from gateways, event/stream monitoring and processing component (SMPC) in the pub/sub broker determines whether it needs processing or just store for periodic send or for immediate delivery.
③ If sensor data needs periodic/ emergency delivery, the analyzer determines which SaaS applications the events belong to and then passes the events to the disseminator along with application ids.
④ The disseminator, using an event matching algorithm, finds appropriate subscribers for each application and delivers the events for use.

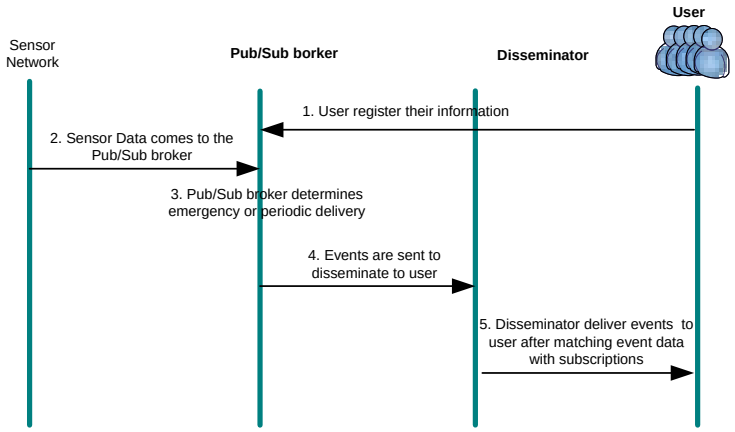


Figure 25 Sequence diagram of pub/sub components workflow

Benefits of our proposed WSN integration mechanism include:

The set of publishers (SA devices) and subscribers (consumer or applications) can dynamically change over time. Consumer or applications do not need to be aware of the failures and changes; they just receive their data when the new devices begin to operate. The same applies to the SA publisher nodes: they do not need to know which applications are interested in their data. They just send their data to the broker, which will then take care of the data distribution to the applications.
For application developers, the pub/sub system hides the complexity of the underlying network and lets them concentrate on the design of the application itself.
To receive data of a certain SA device, the only thing need to know is the content the SA device publishes along with metadata. Even if a SA device is moved to another WSN (e.g., because of network congestion), no change needs to be done to the applications and gateways as long as the SA device is still providing the same content.

Chapter 7

SENSOR DATA DISSEMINATION TO CLOUD

7 Sensor Data Dissemination Mechanism

7.1 Introduction

Sensor data dissemination mechanism is to deliver published sensor data or events to appropriate users or applications from Sensor-Cloud, the *disseminator* component of the *content-based pub/sub broker* needs to match published events with subscriptions efficiently. Designing an efficient content or event matching algorithm is a key challenge in content-based pub/sub system especially for the range predicate case

7.2 Problems of Existing Works

To deliver published sensor data or events to appropriate users or applications from Sensor-Cloud, the *disseminator* component of the *content-based pub/sub broker* needs to match published events with subscriptions efficiently. Designing an efficient content or event matching algorithm is a key challenge in content-based pub/sub system especially for the range predicate case. It is difficult to construct an effective index for multidimensional range predicates. It is even more challenging if these predicates are highly overlapping. In U-Life care application scenario, doctors and caregivers may express their interests into a range (i.e. 35<body temperature < 37). Also multiple event matching is required.

In the literature, there are two major classes of solutions in content-based pub/sub system to match the events with subscriptions: cluster-based approaches and exact match approaches. Various clustering schemes were studied in [42], [43] and recently in [44]. Clustering techniques include spanning tree based clustering, K-means clustering and Grid Partitioning. Clients subscribe to all clusters that overlap, possibly partially, with their interests, and thus may receive false negative events (an event that a subscriber was expected to receive but it has not received) or false positive events (unwanted events). Therefore, the fine-grained expressiveness of content-based pub/sub systems has to be sacrificed to certain degrees in this “clustering” approach. The second category achieves precise content or event delivery: event is only delivered to the clients whose subscriptions match the attribute descriptions of the event [45]-[47]. This “exact-match” approach retains the desirable features of expressiveness and flexibility in content-based pub/sub systems, at the expense of potentially higher state maintenance and processing cost.

The clustering approaches, described above, are not suitable in our application

scenario, since they produce false positive or false negative events which are not acceptable. The exact match approaches are feasible in our case but are not appropriate, since they require huge processing time and thus make delay of delivering events to subscribers.

7.3 Proposed Solution

Statistical Group Index Matching

We propose an event matching algorithm called SGIM (Statistical Group Index Matching) algorithm for the content-based pub/sub broker system that supports single or range predicates or overlapping predicates in subscriptions and provides single or composite event matching. Here, events and subscriptions can be specified in range values. It combines the advantage of clustering and exact match approaches to minimize both the amount of matching inside the pub/sub system and the delay to receive subscribed content at clients. It maintains the flexible expressions of predicates, guarantees the scalability with respect to the number of subscriptions and published events and explicitly addresses the system adaptability problem.

SGIM algorithm operates in two stages. In first stage it preprocesses subscriptions by grouping them by the predicates corresponding to the relevant properties of events, so that for a given event, it can quickly eliminate large amounts of non matching subscriptions and focus on a small subset of possibly matching subscriptions. The grouping idea from statistics is used here. In the second stage, matching subscriptions are derived sequentially. All predicates stored in the system are associated with a unique ID. Similarly subscriptions are identified with subscription id.

Let *S* is a set of subscriptions, $S = \{s_1, s_2 .... s_{n-1}, s_n\}$ where n is the total no. of subscriptions and P is a set of predicates in *S,* $P = \{p_1, p_2 .... p_{n-1}, p_m\}$ where m is the total no. of predicates in a subscription. We will now group the subscriptions using these predicates. In our system, we have two predicates in a subscription (i.e. $p_1 = data > mPV$ and $p_2 = data < MPV$) and these two predicates are used to group the subscriptions. We define a set S' that contains all the subscriptions of S sorted by p_1 value in ascending order. Then we define a grouping sequence $(mG_1, mG_2, ...mG_g)$ such that $g = \lfloor 1 + \log_2(n) \rfloor$ where g is the total no. of groups and n is the total no. of subscriptions in S'. The grouping space, denoted by $SP(S', g)$, is defined as the set containing all such group sequences over S' and g . Then we find an equivalent interval of each group by $IN = \lceil (p_{highest} - p_{lowest}) / g \rceil$ where $p_{highest}$ and p_{lowest} is the highest and lowest value of p_1 .

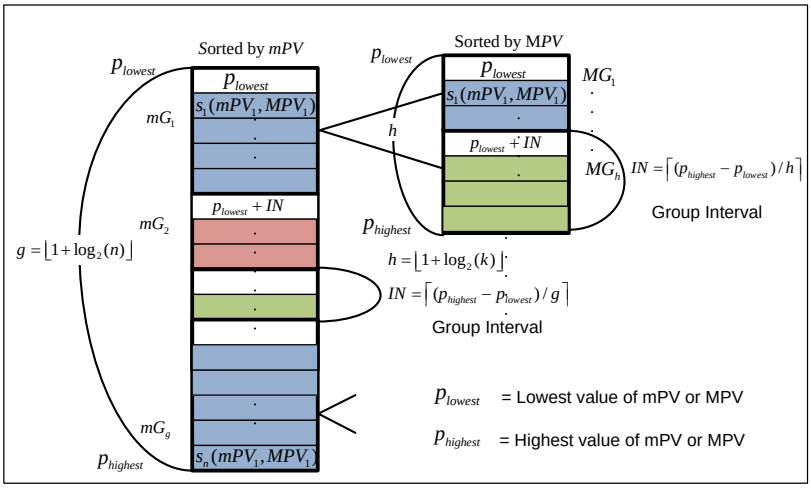


Figure 26 Equivalent interval grouping method of *SGIM*

Again from each $mG_{i=1...g} \in SP(S', g)$, we define another grouping sequence $(MG_1, MG_2, ...MG_h)$ which are sorted by p_2 value in ascending order such that $h = \lfloor 1 + \log_2(k) \rfloor$ and $\sum_{t=1}^{h} MG_t = mG_i$, for any $mG_{i=1..g} \in SP(S', g)$. Here k is the total number of subscriptions in any group of $mG_{i=1..g} \in SP(S', g)$. Then another equivalent group interval is created over $SP(mG_{i=1..g}, h)$ by $\lceil (p_{highest} - p_{lowest}) / h \rceil$ where $p_{highest}$ and p_{lowest} is the highest and lowest value of p_2. Figure 26 illustrates the *SGIM* algorithm's grouping method.

After grouping of subscriptions in the above way, when event comes to the system, it is first matched with the group indexes of $mG_{i=1..g} \in SP(S', g)$ based on the predicate condition and if any match found then compared with group indexes of $MG_{i=1..h} \in SP(mG_1, h)$ and so on. In this way all groups are found that matches with event data. Finally sequential matching is done in the selected groups to find the subscriptions that are satisfied by all predicates in the event. In sequential matching process, when any predicate in a subscription is evaluated to false, there is no need to evaluate the remaining predicates of that subscription. Furthermore, if this predicate is shared by many subscriptions, all remaining predicates in these subscriptions can be ignored, which can significantly reduce the evaluation overhead.

Advantages of the SGIM Algorithm

It supports single, range or overlapping predicates in the subscription and allow multiple or composite event data matching. Events can also be specified in range values.
Grouping and sequential matching mechanism can significantly reduce the matching cost at the pub/sub broker system and thereby improve the delay of content delivery.

These techniques are more beneficial for larger groups
It maintains the flexible expressions of predicates and guarantees the scalability with respect to the number of subscriptions and published events.

Chapter 8

DYNAMIC CLOUD COLLABORATION

8 Dynamic Cloud Collaboration Mechanism

8.1 Introduction

Currently interoperability and scalability are two major challenging issues for Cloud computing. As consumers of different Cloud applications rely on Cloud Providers (CLP) to supply all their computing needs (process, store and analyze huge sensor data and user generated data) on demand, they will require specific QoS to be maintained by their providers in order to meet their objectives and sustain their operations. Forming a dynamic collaboration (DC) platform among Cloud providers (CPs) can help to better address these issues. A DC platform can facilitate to reduce expenses, avoid adverse business impact and to offer collaborative or portable Cloud services to consumers. However, there are two major challenges involved- one is to find an appropriate market model to enable the DC platform and another one is to minimize the conflicts among providers that may happen in a market-oriented DC platform when negotiating among providers [49-50]. We address these two problems and provide candidate solutions.

8.2 Problems of Existing Works

Very few researches have been done regarding the Cloud market. Existing combinatorial auction (CA) based market mechanisms [53] can be utilized but it is not fully suitable to meet the requirements of DC in Cloud market model. In existing CA-based market model, after winning the auction, the winning bidders need to collaborate with each other to enable the DC platform but a large number of conflicts happen when negotiating among providers due to DC. The algorithms in [49] and [50] can be utilized to handle conflicts from happening. But the main problem of these approaches is that auctioneer may choose an improper set of resource providers so conflicts cannot be prevented from happening.

To address the issue of conflicts minimization, CPs can choose suitable partners in a DC platform. However, collaborator or partner selection problem (PSP) is a complex problem, which has been proved to be NP-hard. Also PSP for CPs in the CACM model is different from other PSP problems in areas like manufacturing, supply chain or virtual enterprise [63]-[66] since a large number of conflicts may happen among CPs due to DC. In the existing studies on partner selection, the individual information (INI) is mostly used, but the past collaborative relationship information (PRI) [67] between partners, is overlooked. In fact, the success of past relation between participating CPs may reduce uncertainty and conflicts, short adaptation duration, and also help to the

performance promotion. So the existing methods cannot be applied directly to solve the PSP problem of CPs.

8.3 Proposed Solution

- A novel combinatorial auction (CA) based Cloud market model called CACM with a new auction policy is proposed that facilitates a virtual organization (VO) based dynamic collaboration platform among CPs. To address the issue of conflicts minimization among providers, the new auction policy in CACM model allows a CP to dynamically collaborate with suitable partner CPs to form a group before joining the auction and to publish their group bids as a single bid to fulfill the service requirements completely, along with other CPs, who publishes separate bids to partially fulfill the service requirements. This new approach can create more chances to win the auctions for the group since collaboration cost, negotiation time and conflicts among CPs can be minimized.

- To find a good combination of CP partners required for making groups and reducing conflicts, a multi-objective (MO) optimization model of quantitatively evaluating the partners using their individual information (INI) and past collaborative relationship information (PRI) is proposed. To solve the MO optimization model for partner selection, a multi-objective genetic algorithm (MOGA) that uses INI and PRI called MOGA-IC is also presented as the model is NP-hard. A numerical example is also presented to illustrate the proposed MOGA-IC.

The proposed CACM model to enable DC among CPs is shown in Figure 29.The CACM model allows bidders to make groups and submit their bids for a set of services to auctioneer as a single bid while also supporting the bidder to submit bid separately for a set of services. The Figure 27 and Figure 28 illustrate this scenario. We use the auction scheme based on [53] to address the CACM model. We use eContract service [51] to capture the contributions as well as agreements among all service providers for dynamic collaboration.

We define the CACM model in which the main participants are brokers, users/consumers, Cloud resource/service providers, and trustworthy auctioneers. For submitting group bids as a single bid, the interested CPs dynamically collaborate with other CPs and make groups by using a partner selection approach. They make initial agreement (soft contract) using eContract among each other and submit group bids as a single bid. If this group wins the auction for a set of services, they finalize the agreement (hard contract) and provide composite or collaborative services to consumers. On the other hand, if CPs win the auction separately for each service, they need to collaborate (agree with resources contributed by other provider) with each

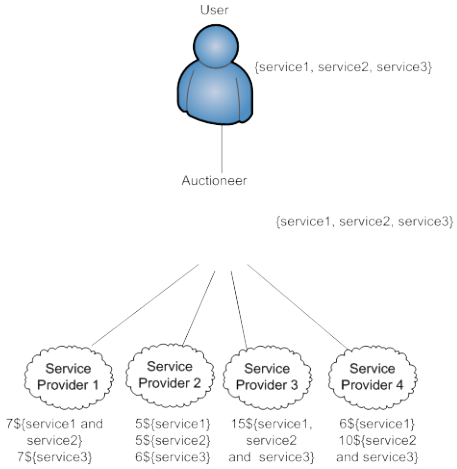


Figure 27 Existing Combinatorial Auction

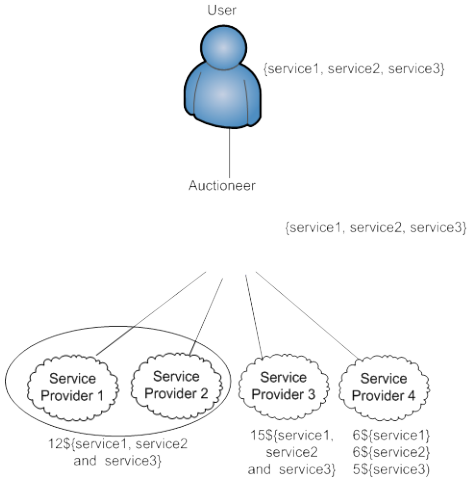


Figure 28 Collaborative Combinatorial Auction

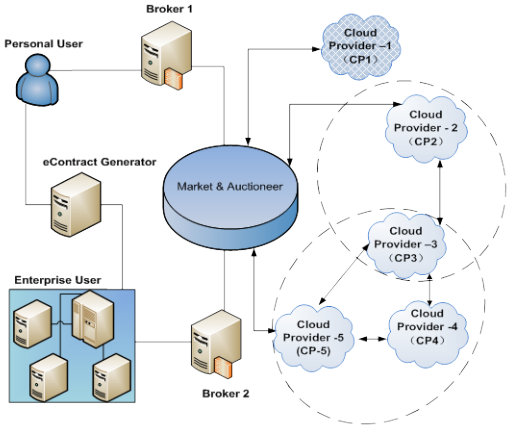


Figure 29 Architecture of our proposed auction-based Cloud market model

System Model for Auction in CACM

For the convenience of analysis, the parameters and variables for the auction models are defined as follows:

$R = \{R_j \mid j = 1 \ldots n\}$: a set of n service requirements of consumer where $S \subseteq R$

$P = \{P_r \mid r = 1 \ldots m\}$: a set of m Cloud providers who participate in the auction as bidders where $G \subseteq P$

P_{rj} = a Cloud provider r who can provide service j

$S(P_r) =$ a set of services $(S_{j=1 \ldots n})$ provided by any CP *r* where $S(P_r) \subseteq R$

$\Omega_{\max}(R,Q)$ = payoff function of the user where *R* is the service requirements and *Q* defines SLAs of each service.

Single and Group Bidding Functions of CPs

Let M be the service cost matrix of any CP P_r. We assume that each CP can provide at most two services. The matrix *M* includes costs of P_r provider's own services as well as the collaboration costs (CC) between services of its own and other providers. Figure 29 illustrates the matrix M. We assume that P_r provides two services - CPU and Memory. Let $a_{ii}(i=1...n)$ be the cost of providing any service in *M* independently, a_{ij} $(i,j=1...n, i \neq j)$ be the CC between S_i and S_j services ($S_i, S_j \in S(P_r)$) and $a_{ik}(i,k=1...n, i \neq k)$ be the CC between S_i and S_k services ($S_i \in S(P_r)$ and $S_k \notin S(P_r)$). We set nonreciprocal CC between $S(P_r)$ services in *M* which is practically reasonable. If CP P_r knows other providers or have some past collaboration experience with others, it can store true CC of services with other providers. Otherwise it can set a high CC for other providers. The CC of services with other providers in matrix *M* is updated when the providers finish negotiation and collaboratively provide the services of consumers in the DC platform.

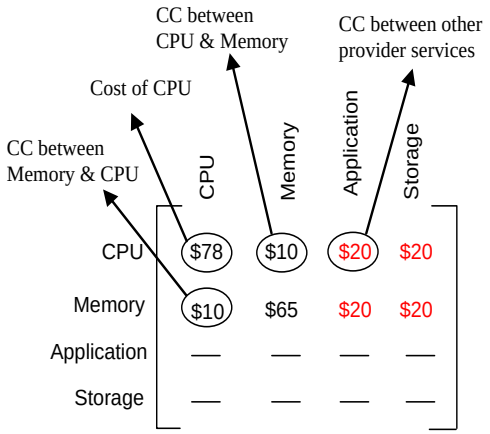


Figure 30 Cost Matrix M

Now the *Bidding Function* of any CP say P_r who submits bid separately to partially fulfill the customer service requirements without collaborating with other CPs can be determined as follows: $\phi_{S(P_r)} = C_{S(P_r)} + \gamma(P_r)$ where $C_{S(P_r)}$ is the total cost incurred by CP P_r to provide $S(P_r)$ services $(S(P_r) \subseteq R)$ and $\gamma(P_r)$ is the expected profit of provider P_r. The total cost $C_{S(P_r)}$ is calculated as follows by using the matrix M :

$$C_{S(P_r)} = \sum_{S_i \in S(P_r)} a_{ii} + \sum_{S_i \in S(P_r)} \sum_{S_j \in S(P_r)} a_{ij} + \sum_{S_i \in S(P_r)} \sum_{S_k \notin S(P_r)} a_{ik} \qquad (1)$$

where, $i, j, k = 1...n$ and $i \neq j \neq k$

The first term in the equation (1) is the cost of providing services $S(P_r)$ .The second term is the total collaboration cost between $S(P_r)$ services and third term refers to the total collaboration cost between services of different CPs with whom provider P_r needs to collaborate. As provider P_r does not know to whom it will collaborate after winning the auction, the true cost of a_{ik} cannot be determined. Therefore, P_r may set a high collaboration cost in a_{ik} in order to avoid potential risk in collaboration phase.

Now the *Bidding Function* of a group of CPs, who submit their bids collaboratively as a single bid to fulfill the service requirements completely, can be determined as follows: Let P_r forms a group G by selecting appropriate partners where $S(P_G)$ be the set of services provided by G and $S(P_G) \subseteq R, G \subseteq P$. For any provider like $P_r \in G$, the total cost of providing $S(P_r)$ services is:

$$C^G_{S(P_r)} = \sum_{S_i \in S(P_r)} a_{ii} + \sum_{S_i \in S(P_r)} \sum_{S_j \in S(P_r)} a_{ij} + \sum_{S_i \in S(P_r)} \sum_{S_g \in S(P_G) \setminus S(P_r)} a_{ig} + \sum_{S_i \in S(P_r)} \sum_{S_k \notin S(P_G)} a_{ik}$$

where, $i, j, g, k = 1...n$ and $i \neq j \neq g \neq k$

We can see from equation (2) that the term $\sum_{S_i \in S(P_r)} \sum_{S_k \notin S(P_r)} a_{ik}$ of eq. (1) is now divided into two terms in $C^G_{S(P_r)}$: $\sum_{S_i \in S(P_r)} \sum_{S_g \in S(P_G) \setminus S(P_r)} a_{ig}$ and $\sum_{S_i \in S(P_r)} \sum_{S_k \notin S(P_G)} a_{ik}$. The term $\sum_{S_i \in S(P_r)} \sum_{S_g \in S(P_G) \setminus S(P_r)} a_{ig}$ denotes the total collaboration cost of services of provider P_r with other providers in the group. The term $\sum_{S_i \in S(P_r)} \sum_{S_k \notin S(P_G)} a_{ik}$ refers to the total collaboration cost between services of other CPs outside of the group with whom provider P_r needs to collaborate. This term can be zero if the group can satisfy all the service requirements of consumer. Since P_r knows other group members, it can find the true value of the term $\sum_{S_i \in S(P_r)} \sum_{S_g \in S(P_G) \setminus S(P_r)} a_{ig}$. Moreover, if P_r applies any good strategy to form the group G , it is possible for P_r to minimize $\sum_{S_i \in S(P_r)} \sum_{S_g \in S(P_G) \setminus S(P_r)} a_{ig}$. Hence, this group *G* has more chances to win the auction as compare to other providers who submit separate bids to partially fulfill the service requirements. So the *Bidding Function* for the group *G* can be calculated as follows:

$$\phi^G_{S(P_G)} = \sum (C^G_{S(P_r)} + \gamma^G(P_r)), \forall P_r \in G, r = 1...l$$

where *l* is the no. of providers in *G* and $\gamma^G(P_r)$ is the expected profit of any provider *r* in the group.

Payoff Function of the User/Consumer

With the help of broker user generates the payoff function. During auction, user uses the payoff function $\Omega_{\max}(R,Q)$ to internally determine the maximum payable amount that it can spend for a set of services. If the bid price of any CP is greater than the maximum payable amount $\Omega_{\max}$, it will not be accepted. In the worst case, auction terminates when the bids of all Cloud service provider is greater than $\Omega_{\max}$. In such case, user modifies its payoff function and the auctioneer reinitiates auction with changed payoff function.

Profit of the CPs to form a Group

Let $\phi^{G}_{S(P_r)}$ be the price of the provider r when it forms a group *G* where $C^{G}_{S(P_r)}$ is the cost of its services in the group. So the expected profit for the P_r in the group is $\gamma^{G}(P_r)=\phi^{G}_{S(P_r)}-C^{G}_{S(P_r)}$. We know that the expected profit for the provider r, who submits bid separately, is $\gamma(P_r)=\phi_{S(P_r)}-C_{S(P_r)}$. We argue that if any CP forms a group using a good partner selection strategy; it can increase its profit rather than separately publishing the bid. To calculate the increased profit, we consider the following assumptions:

$$C^{G}_{S(P_r)} \le C_{S(P_r)} \text{ and } \gamma^{G}(P_r)=\gamma(P_r)$$

Since CP *r* can collaboratively publish the bid, it may minimize its collaboration cost by selecting good partners, that is, $C^{G}_{S(P_r)}$ should be less than or equal to $C_{S(P_r)}$. However, $\gamma^{G}(P_r)=\gamma(P_r)$ means the expectation of profit does not change. Consequently, we can also deduce this: $\phi^{G}_{S(P_r)} \le \phi_{S(P_r)}$ .That is the provider who collaboratively publishes bid can provide lower price for its services while maintaining the same expected profit. Thus it has more chances to win the auction. To determine the increased profit for P_r, let $\phi^{2\text{LP}}_{S(P_r)}$ be the second lowest price that will be paid to P_r for $S(P_r)$ services if it wins the auction. Now if P_r attends any auction and apply separate and collaborative bidding strategy alternatively, the increased profit $\gamma^{I}(P_r)$ for P_r can be calculated as follows:

$$\gamma^{I}(P_r)=\alpha(\phi^{2LP}_{S(P_r)}-C^{G}_{S(P_r)})-\beta(\phi^{2LP}_{S(P_r)}-C_{S(P_r)})$$

where

$$\alpha=\begin{cases}1 & \text{if provider } r \text{ collaboratively wins the auction}\\ 0 & \text{otherwise}\end{cases} \quad (5)$$

$$\beta=\begin{cases}1 & \text{if provider } r \text{ separately wins the auction}\\ 0 & \text{otherwise}\end{cases}$$

From equation (5), we can figure out that if provider *r* collaboratively wins the auction, it can always get the increased profit. Otherwise, no increased profit will be achieved. So a good partner selection strategy is required for a CP to make groups. In

the next section, we will describe an effective MO optimization model for a good combination of partner selection.

MO Optimization Model for Partner Selection

A primary/initiator CP (pCP) identifies a business opportunity which is to be addressed by submitting a bid for a set of services for the consumer. It needs to dynamically collaborate with one or more partners to complete the requirements of the consumer as it cannot provide all the services. We assume that each CP can provide one or at most two services and each service has one or more providers. Also each CP can organize other groups simultaneously. The pCP has the individual and collaborative information of all the other providers for each service. Individual information includes price and quality of service information of other providers. Collaborative or relationship information includes number of projects/auctions accomplished/won by other providers among themselves and also with pCP. The pCP can get all of these information from each CPs website, market and also from customer feedback. Let,

ϕ_{rj} = the price of CP *r* for providing service *j* independently

Q_{rj} = the quality value for service *j* of CP *r* (qualitative information can be expressed by the assessment values from 1 to 10 (1: very bad, 10: very good))

$W_{rj,xi}$ = the value of past collaboration experience (i.e. number of times collaboratively win the auctions) between provider *r* for service *j* and provider *x* for service *i* where $(r, x = 1....m; i, j = 1....n; i \neq j)$

$U = \{U_{rj} \,|\, r = 1.....m,\ j = 1....n\}$: a decision vector of partner selection.

To resolve the partner selection problem of a pCP using the INI and PRI, a multi-objective (MO) optimization model to minimize total price and maximize total collaborative past relationship (PR) performance and service quality values can be expressed mathematically as follows:

$$\text{Minimize Obj\_1} = \sum_{j=1}^{n}\sum_{r=1}^{m} \phi_{rj} U_{rj}$$

$$\text{Maximuze Obj\_2} = \sum_{j=1}^{n}\sum_{r=1}^{m} Q_{rj} U_{rj}$$

$$\text{Maximize Obj\_3} = \sum_{\substack{i,j=1 \\ i \neq j}}^{n}\sum_{r,x=1}^{m} W_{rj,xi} U_{rj} U_{xi}$$

subjec to

$$U_{rj} = \begin{cases} 1 & \text{if choose } P_{rj} \\ 0 & \text{otherwise} \end{cases}$$

$$U_{rj}U_{xi} = \begin{cases} 1 & \text{if choose } P_{rj} \text{and } P_{xi} \\ 0 & \text{otherwise} \end{cases}$$

Multi-Objective Genetic Algorithm

In this section, the proposed MOGA-IC is designed to solve the MO optimization model of CP partner selection as follows:

The natural number encoding is adopted to represent the chromosome of individual. A chromosome of an individual is an ordered list of CPs. Let $y = [y_1, y_2, ..., y_j .....y_n\ ]\ (j = 1, 2, ... n), y_j$ is a gene of the chromosome, its value is between 1 and *m* (for service *j*, there are *m* CPs to response). If $m = 50$ and $n = 5$, there are *10* CPs who can provide each service j. So total 10^5 possible solutions available. In this way the initial populations are generated. For the selection of individual, the binary tournament selection strategy is used. We employ two-point crossover. In case of mutation, randomly one provider is changed for any service.

The multi-objective functions (Obj_1, Obj_2 and Obj_3) are considered as fitness functions when calculating the fitness values. A fast non-dominated sorting approach based on NSGA-II [12] is employed to calculate the fitness values of individual. Any two individuals are selected and their corresponding fitness values are compared according to the dominating-relationships and crowding-distances in the objective space. Then all the individuals are separated into the non-dominated fronts. The individuals in the same fronts do not dominate each other and we call this non-dominated sorting. Now the MOGA is presented step by step as follows:

Step 1: Initialize the input parameters which contain the number of requirements (*R*), providers (*m*) and maximum genetic generations (*G*), population size (*N*), crossover probability (p_c) and mutation probability (p_m).

Step 2: Generate the initial parent population $P_t, (t = 0)$ of size N_P.

Step 3: Apply binary tournament selection strategy to the current population, and generate the offspring population O_t of size $N_O = N_P$ with the predetermined p_c and p_m.

Step 4: Set $S_t = P_t \cup O_t$, apply a non-dominated sorting algorithm and identify different fronts $F_1, F_2 ... F_a$.

Step 5: If the stop criterion ($t > G$) is satisfied, stop and return the individuals (solutions) in population P_t and their corresponding objective values as the Pareto-(approximate) optimal solutions and Pareto-optimal fronts.

Step 6: Set new population $P_{t+1} = 0$. Set counter $i = 1$. Until $|P_{t+1}| + |F_i| \le N$ set $P_{t+1} = P_{t+1} \cup F_i$ and $i = i + 1$.

Step 7: Perform the crowding-sort procedure and include the most widely spread $\left(N - |P_{t+1}|\right)$ solutions found using the crowding distance values in sorted F in P_{t+1}.

Step 8: Apply binary tournament selection, crossover and mutation operators to P_{t+1} to create offspring population O_{t+1}.

Step 9: Set $t = t + 1$, then return to Step 4.

Simulation Results

In this section, we present a simulation example of PSP for a pCP in the CACM model. It is used to illustrate the proposed MOGA-IC method. NSGA-II is utilized to develop the MOGA-IC. We implement the CACM model (winner determination algorithm) with new auction policy as well as the MOGA-IC in Visual C++.

One of the main challenges in the CACM model and the PSP of CP is the lack of real-world input data. So we conduct the experiments using synthetic data. We generate the input data as follows:

Many CPs (*m* = 100) with different services and also some consumer requirements (*R* = 3-10) are generated randomly. We assume that each CP can provide at most 2 services so that they have to collaborate with others to fulfill the service requirements *R*. Each service may have one or more CPs. Based on *R*, CPs are selected. So it is possible that every CP may not provide the required *R*. Also the cost of providing any independent service is randomly generated from \$80 to \$100. The ranges of collaboration cost (*CC*) of services as well as the profit are set within \$10 - \$30 and \$10 - \$20 respectively. Quality and collaborative performance values (number of auctions collaboratively won by other providers among themselves and also with pCP) of providers are randomly selected from 1-10 and 0-10 respectively. If any provider has more collaboration experience with other providers, the *CC* can be minimized. We use the following formula to calculate the *CC* between any provider P_{rj} and P_{xi} :

$$CC_{rj,xi} = CC_{\min} + (CC_{\max} - CC_{\min}) \times \frac{1}{e^{W_{rj,xi}}}$$

where

$CC_{\min}$ = the minimum CC between services (here \$10)

$CC_{\max}$ = the maximum CC between services (here \$30)

$W_{rj,xi}$ = the value of number of collaboration experience between P_{rj} and P_{xi}. If it is zero, the highest CC is set between providers. Thus the final price of services is generated for each provider and it is varied based on CC in different auctions. Also the individual and collaborative information are normalized using the method proposed by Hwang and Yoon [8].

Assume that there are five service requirements (i.e. R = 5) in the auctions. To provide these services, pCP needs to find appropriate partners and to submit bids collaboratively as single bid. We also assume that individual information and collaborative information about candidate partners (e.g. m = 50) are available to the pCP. The parameters for the proposed MOGA are set as follows: $N = 50$, $G = 50$, $p_c = 0.9$ and $p_m = 0.1$. The algorithm is executed three times and the average running time is 0.0361 seconds. The first 16 Pareto-optimal solutions of the first front of MOGA-IC are presented in Table 4 and graphically presented by Figure 31. The pCP

can select any combination of CP partners from the solutions.

Also from Figure 32, we can see the average optimized values of three objective functions in the first fronts during 50 generations. The simulation experimental results show that the proposed method can support satisfactory and high quality partner selection.

Table 4: Pareto-optimal solutions of MOGA-IC with NSGA-II for example 1

Pareto-optimal Solutions	Optimal Objective Function Values		
$y = (y_7 \quad y_4 \quad y_3 \quad y_2 \quad y_8)$	Obj_1	Obj_2	Obj_3
1 18 6 10 32	3.16	3.32	7.63
1 18 10 10 32	3.49	4.2	7.33
1 34 10 28 32	2.94	4.35	6.24
1 9 32 28 21	1.37	3.66	4.93
1 9 10 28 32	2.44	4.23	6.65
1 9 32 28 19	1.45	3.81	5.49
1 9 14 28 32	2.00	3.58	6.82
1 34 32 10 32	2.23	4.01	6.97
1 18 10 28 32	3.28	4.31	6.44
1 9 32 10 32	1.73	3.89	5.88
1 34 32 28 32	2.02	4.12	5.59
1 34 6 10 32	2.82	3.36	7.39
1 34 32 10 19	2.16	3.82	6.48
1 34 10 10 32	3.15	4.24	7.12
1 9 32 28 32	1.52	4.00	5.44
1 18 14 10 32	3.05	3.55	7.27

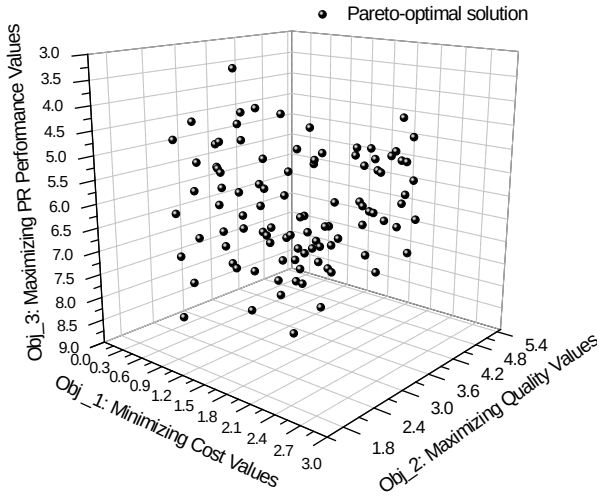


Figure 31 Pareto-optimal solutions of MOGA-IC (N = 50 and G = 50) obtained by NSGA-II

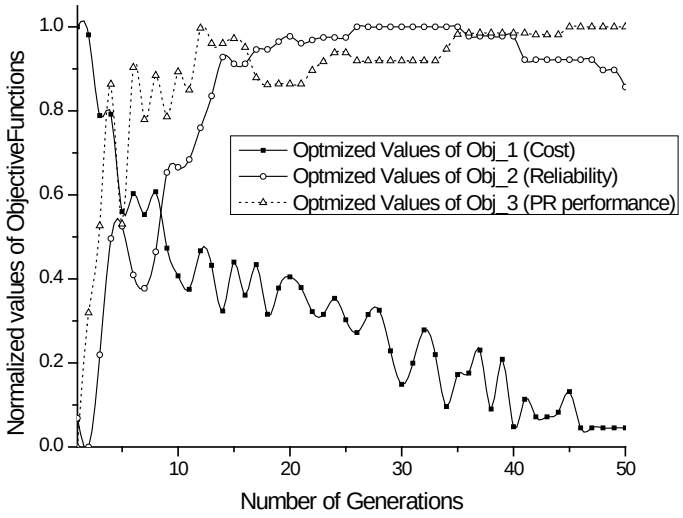


Figure 32 Three objective functions are optimized during each generation

❖ ***Benefits of Our Approach***

Minimize negotiation time

Fewer conflicts exist among Cloud service providers

Providers in the group can provide lower service cost for consumer and thus have higher chance to win the bid than single providers

Chapter 9

ACTIVITY RECOGNITION ENGINE FOR U-LIFE CARE

9 Activity Recognition Engine for u-Life Care Services

9.1 Introduction

One of the main targeting services of u-Life care is to enable people to live independently longer through the early detection and prevention of chronic disease and disabilities. Computer vision, emplaced wireless sensor networks (WSN), and body networks are emerging technologies that promise to significantly enhance medical care for seniors living at home in assisted living facilities. With these technologies, we can collect video, physiological, and environmental data, identify individuals' ***activities of daily living*** (ADL), and act for improved daily medical care as well as real-time reaction to medical emergencies. Overall, projected benefits include greater independence for the elderly, lower medical costs through reduction in hospital and emergency room visits, improved health, and via longitudinal studies, increased understanding of the causes of diseases and the efficacy of their treatment.

To achieve that, accurately identifying individuals' ADL, so-called ***activity recognition*** (AR) which can be based on both video and sensor (e.g., accelerometer, gyroscope, physiological) data, is of vital importance. However, it is a significant challenge. Video-based AR can be complex due to abrupt object motion, noise in images, no n-rigid or articulated nature of human body, partial and full object occlusions, scene illumination changes, and real-time processing requirements, etc. Sensor-based AR involves the process of automatically classifying raw streams of sensor data that requires a sophisticated set of smoothing, filtering, and feature extraction algorithms for analog sensors and the ability to deal with false positives or negatives from digital sensors. The process is further complicated because a person will not perform an activity in the same way each time; the user may often perform multiple activities simultaneously; and every individual will perform activities differently.

9.2 Problems of Existing Works

The practical benefits, the scientific complexity, and the speed and price of curre nt hardware have intensified the effort within the research community toward s automatic capture and analysis of human activity. This is evident by looking at the number of publications, special sessions/issues at the major conferenc e/journals as well as the number of workshops directly devoted to such topic s. However, in video-based AR, ***progress can only be seen for recognition of simple activities such as walking, running, and sitting***. Here, only a small bo dy of literature goes beyond these simple activities into motion interpretation

where scene context and the interaction with other humans are considered, e.g., [56]-[62]. Much more work is required before AR is reliable enough for u-Life care.

In sensor-based AR, many research groups have been investigating how to construct an efficient system. Researchers at Intel Research Seattle and the University of Washington have built a prototype system that can infer a person's ADLs. Sensor tags are placed on everyday objects such as a toothbrush or coffee cup. University of Rochester is building The Smart Medical Home, which is a five-room house outfitted with infrared sensors, computers, bio-sensors, and video cameras for use by research teams to work with research subjects as they test concepts and prototype products. Georgia Tech built an Aware Home as a prototype for an intelligent space. Massachusetts Institute of Technology (MIT) and TIAX are working on the PlaceLab initiative, which is a part of the House_n project. The mission of House_n is to conduct research by designing and building real living environments -``living labs''- that are used to study technology and design strategies in context. Many projects are building body networks for the collection of vital signs, such as AMON. All these systems demonstrate the excitement and need for such systems.

9.3 Proposed Solution

In our proposed work, we will consider video data and body-attached sensor data home health care and assisted living. While many research groups have been addressing the problem of determining ADL, most focus on one or a few activities, use one or a few techniques and often lack robustness to determine those activities in difficult situations. Our research aims to improve the robustness and scope of ADL capabilities. In particular, we propose:
To improve the accuracy of video-based AR by developing novel algorithms for human body detection and motion feature extraction,
To …. (sensor-based), and
To … (ontology).

9.3.1 Proposed algorithm for human body detection

The accuracy of the video-based AR depends significantly on the performance of human body segmentation. In the field of image segmentation, since it was first introduced by Kass et al. [61][61] in 1988, active contour (AC) model has attracted much attention. Recently, Chan and Vese (CV) proposed in [62] a novel form of AC based on the Mumford and Shah functional for segmentation and the level set framework. Unlike other AC models which rely much on the gradient of the image as the stopping term and thus have unsatisfactory performance in noisy images, the CV AC model does not use the edge information but utilizes the difference between the regions inside and outside of the curve, making itself one of the most robust and thus widely used techniques for image segmentation. Its energy functional is defined by

$$F(C) = \int_{in(C)} \left| I(\mathbf{x}) - c_{in} \right|^2 d\mathbf{x} + \int_{out(C)} \left| I(\mathbf{x}) - c_{out} \right|^2 d\mathbf{x} \quad (0.1)$$

where $\mathbf{x} \in \Omega$ (the image plane) $\subset R^2$, $I : \Omega \to \mathcal{Z}$ is a certain image feature such as intensity, color, or texture, etc., and c_{in} and c_{out} are respectively the mean values of image feature inside $\left[in(C)\right]$ and outside $\left[out(C)\right]$ the curve C, which represents for the boundary between two separate segments. Considering image segmentation as a clustering problem, we can see that this model forms two segments (clusters) such that the differences within every segment are minimized. However, the global minimum of the above energy functional does not always guarantee the desirable results, especially when a segment is highly inhomogeneous, e.g., human body, as can be seen in Figure 33(b). The unsatisfactory result of the CV AC in this case is due to the fact that it is trying to minimize the dissimilarity within each segment but does not care about the distance between different segments. Our methodology is to propose to incorporate an evolving term based on the Bhattacharyya distance to the CV energy functional such that not only the differences within each region are minimized but the distance between the two regions is maximized as well. The proposed energy functional is

$$E_0(C) = \beta F(C) + (1-\beta) B(C) \quad (0.2)$$

where $\beta \in [0,1]$, $B(C) \equiv B = \int_{\mathcal{Z}} \sqrt{p_{in}(z)\, p_{out}(z)}\, dz$ the Bhattacharyya coefficient with

$$p_{in}(z) = \frac{\int_\Omega \delta(z - I(\mathbf{x})) H(-\phi(\mathbf{x})) d\mathbf{x}}{\int_\Omega H(-\phi(\mathbf{x})) d\mathbf{x}} \quad (0.3)$$

$$p_{out}(z) = \frac{\int_\Omega \delta(z - I(\mathbf{x})) H(\phi(\mathbf{x})) d\mathbf{x}}{\int_\Omega H(\phi(\mathbf{x})) d\mathbf{x}} \quad (0.4)$$

$\phi : \Omega \to R$ the level set function, and $H(\cdot)$ and $\delta(\cdot)$ $H'(\cdot)$ respectively the Heaviside and the Dirac functions~\cite{Chan01}. Note that the Bhattacharyya distance is defined by $\left[-\log B(C)\right]$ and the maximization of this distance is equivalent to the minimization of $B(C)$. Note also that to be comparable to the $F(C)$ term, in our implementation, $B(C)$ is multiplied by the area of the image because its value is always within the interval $[0,1]$ whereas $F(C)$ is calculated based on the integral over the image plane. In general, we can regularize the solution by constraining the length of the curve and the area of the region inside it. Therefore, the energy functional is defined by

$$E(C) = \gamma \int_{\Omega} |\nabla H(\phi(\mathbf{x}))| \, d\mathbf{x} + \eta \int_{\Omega} H(-\phi(\mathbf{x})) d\mathbf{x} + \beta F(C) + (1-\beta) B(C) \tag{0.5}$$

where $\gamma \geq 0$ and $\eta \geq 0$ are constants.

The intuition behind the proposed energy functional is that we seek for a curve which 1) is regular (the first two terms) and 2) partitions the image into two regions such that the differences within each region are minimized (i.e., the $F(C)$ term) and the distance between the two regions is maximized (i.e., the $B(C)$ term).

The level set implementation for the energy functional in (0.6) can be derived as

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| \left\{ \begin{array}{l} \gamma\kappa + \eta + \beta\left[(I - c_{in})^2 - (I - c_{out})^2 \right] \\ -(1-\beta)\left[\frac{B}{2}\left(\frac{1}{A_{in}} - \frac{1}{A_{out}} \right) + \frac{1}{2}\int_{z} \delta(z-I)\left(\frac{1}{A_{out}}\sqrt{\frac{p_{in}}{p_{out}}} - \frac{1}{A_{in}}\sqrt{\frac{p_{out}}{p_{in}}} \right) dz \right] \end{array} \right\} \tag{0.7}$$

where A_{in} and A_{out} are respectively the areas inside and outside the curve C.

As a result, the proposed model can overcome the CV AC's limitation in segmenting inhomogeneous objects as shown in Figure 33(c), yielding the body detector more robust to illumination changes and clothing.

(a) (b) (c)

Figure 33 Sample segmentation of inhomogeneous body-shape object using active contours. (a) Initial contour, (b) result of CV AC, and (c) result of our approach. The CV AC fails to capture the whole body whereas our approach can.

❖ ***Proposed algorithm for motion feature extraction***

After obtaining a set of body silhouettes segmented from a sequence of images (see Figure 34 for an example); we propose to apply ICA (independent component analysis) to get the motion features of that sequence. ICA focuses on the local feature information rather than global information as in PCA (principal component analysis). The extracted features are then symbolized using vector quantization algorithms such

as K-mean or LBG (Linde, Buzo, and Gray) clustering. Finally, symbol sequence is used to generate a codebook of vectors for training/recognizing with HMM (Hidden Markov Model).

Figure 34 Binary silhouettes from a walking sequence.

The overall architecture of proposed system is shown in Figure 35, where T represents the number of testing shape images, N number of trained HMMs, and L likelihoods.

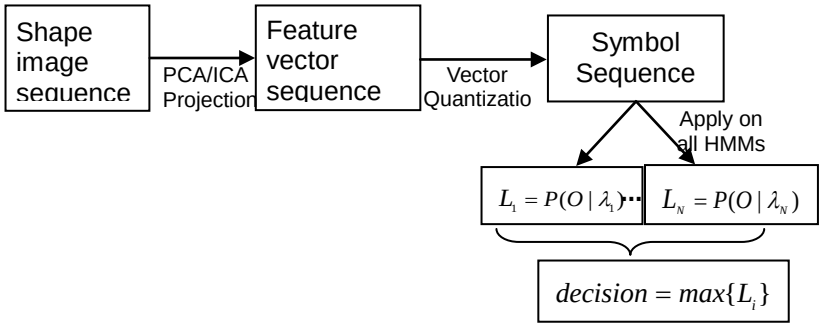


Figure 35 Architecture of the proposed approach for motion feature extraction and recognition.

The system robustness can be further improved using *learned models of pose and motion* which are more general, i.e., able to capture a wide range of h uman movements. The current video-based AR approaches can recognize only a small number of activities (usually simple ones) due to the limited models of pose and motion in 2D images. The 3D body modeling can be used to ge nerate many and more complex models.

Furthermore, sensor-based approaches can help since it is easier for those ap proaches to extend the number of activities to be recognized.

Experiment Results

-
- ***In order to evaluate the proposed video-based activity recognition system, we used a publicly available dataset \cite{Gorelick07}. In this dataset, video clips of nine activities were recorded, namely "bend", "jack" (jumping-jack), "jump" (jumping forward on two legs), "run", "side" (gallopsideways), "skip", "walk", "wave1" (wave-one-hand), and "wave2" (wave-two-hands). Each activity was***

performed by nine different people. Each clip was down-sampled to yield two clips. By this way, we have, for each activity, nine clips for training and nine for testing. The lengths of the clips are not necessarily identical because the input of the HMM can be of various lengths. That is one major advantage of our proposed approach. The video frames were resized to 100 x 70. The recognition rates are summarized in Table 5.

❖

Table 5. Recognition rates for the testing dataset. The average accuracy is 91.4%.

❖

	bend	jack	jump	run	side	skip	walk	wave1	wave2
bend	89%								
jack		89%							
jump			78%						
run				100%					
side					89%				
skip			11%			89%			
walk			11%		11%		100%		
wave1								100%	
wave2									100%
Unknown	11%	11%				11%			

9.3.2 Proposed approach for sensor-based AR

Based on existing work [CITE], we develop our own recognition which is called "semi-Markov Conditional Random Fields (semiCRF)", futhermore we propose a novel algorithm which helps to reduce the complexity of training and inference by more than 10 times in comparison with the original work. In our model, we assume that

$$X = \{x_1, x_2, \ldots, x_T\}$$
$$Y = \{y_1, y_2, \ldots, y_T\}$$

are the input signal and input label respectively. Our goal is to optimize the model parameter so that P(Y|X) is maximized. With conventional conditional random fields P(Y|X) is calculated by:

$$P(Y|X) = \frac{\prod_{t=1}^{T} \Psi(y_{t-1}, y_t, x)}{Z_X}$$

$$\Psi(y_{t-1}, y_t, x) = e^{W^T F(y_{t-1}, y_t, x)}$$

$$Z_X = \sum_{Y'} \prod_{t=1}^{T} \Psi(y'_{t-1}, y'_t, x)$$

Where F is a vector of feature functions (which are often delta functions), W^T is a vector of model parameters, and ψ is called potential functions. Z_X, the normalization factor, is computed by using forward/backward algorithm. However, conventional CRF is limited to Markov assumption therefore it is not able to model the duration of activity as well as long-transition between activities. To overcome these disadvantages we introduce a semi-Markov model by defining a new state as $s_i = (y_i, b_i, e_i)$ where s_i is the i^{th} state, y_i, b_i, e_i in that order are label, beginning time and ending time of the state. For example, given an input label sequence Y=(1,1,2,2,2,3,4,4), then the semi-Markov state sequence is (1,1,2), (2,3,5), (3,6,6), (4,7,8). Note that, in activity recognition we just consider states that have an expected label, because of this we drop unexpected-label states. With these definition, the potential function is rewritten as:

$$\Psi(s_{i-1}, s_i, X) = \begin{pmatrix} e^{Q^T(y_{i-1}, y_i)} \times \\ e^{Q^D(y_i, e_i - b_i + 1)} \times \\ e^{Q^O(y_i, b_i, e_i)} \times \\ e^{Q^O(IA, e_{i-1}+1, b_i - 1)} \end{pmatrix}$$

where

$$Q^T(y', y) = w^T(y', y)\delta(y_{t-1} = y', y_t = y)$$

is a weighted transition potential function, $w^T(y',y)$ is the weight of transition from y' to y.

$$\delta(X) = \begin{cases} 1 \text{ if X is true} \\ 0 \text{ if X is false} \end{cases}$$

$$Q^D(y, d) = w^D(y)\frac{(d - m_y)^2}{2\sigma_y^2}\delta(y_t = y)$$

is a weighted duration potential function of an expected activity, $w^D(y)$ is the weight of duration of activity y, m_y and σ_y are mean and standard deviation respectively.

$$Q^O(y, t_1, t_2) = \sum_{t=t_1}^{t_2} w^O(y, o)\delta(y_t = y, x_t = o)$$

is a weighted observation potential function of an activity y, $w^o(y,o)$ is the observation weight given that input symbol o is observed under label y.

With the above potential function, the likelihood of the training data is given by:

$$P(S|X) = \frac{\prod_{i=1}^{P} \Psi(s_{i-1}, s_i, x)}{Z_X}$$

$$Z_X = \sum_{S'} \prod_{i=1}^{P} \Psi(s'_{i-1}, s'_i, x)$$

Followings are pseudocode to implement forward, backward and Viterbi algorithm.

Forward

for $t = 1$ *To* T **do**
  for $y = 1$ *To* $StateNum$ **do**
    $\alpha[y][t] = 0;$
    $\gamma[y][t] = 0;$
    $\lambda[y][t] = 0;$
    for $d = 1$ *To* D **do**
      if $t - d + 1 > 0$ **then**
        $\gamma[y][t]$ += $\lambda[y][t-d]e^{G(y,t-d+1,t)}$
        $\gamma[y][t]$ += $e^{Q^O(IA,0,t-d)+G(y,t-d+1,t)}$
      else
        Break
    if $t > 1$ **then**
      $\alpha[y][t] = \alpha[y][t-1]e^{Q^O(IA,t,t)} + \gamma[y][t]$
    else
      $\alpha[y][t] = \gamma[y][t]$
    for $y' = 1$ *To* $StateNum$ **do**
      $\lambda[y][t] = \lambda[y][t] + \alpha[y'][t]e^{Q^T(y',y)}$

end

Backward

for $t = T$ *Down To* 1 **do**
  for $y = 1$ *To* $StateNum$ **do**
    $\beta[y][t] = 0;$
    $\eta[y][t] = 0;$
    $\zeta[y][t] = 0;$
    for $d = 1$ *To* D **do**
      if $t + d - 1 \leq T$ **then**
        $\eta[y][t]$ += $\zeta[y][t+d]e^{G(y,t,t+d-1)}$
        $\eta[y][t]$ += $e^{G(y,t,t+d-1)+Q^O(IA,t+d,T)}$
      else
        Break
    if $t < T$ **then**
      $\beta[y][t] = \beta[y][t+1]e^{Q^O(IA,t,t)} + \eta[y][t]$
    else
      $\beta[y][t] = \eta[y][t]$
    for $y' = 1$ *To* $StateNum$ **do**
      $\zeta[y][t] = \zeta[y][t] + \beta[y'][t]e^{Q^T(y,y')}$

end

Step 1. Initialization

$y^* = \text{argmax}\ \delta(y, T)$

$t = T$

$y = y^*$

$i = 1$

end

Step 2. Backtracking

while $y \neq IA$ *and* $t > 0$ **do**

while $\Delta^{Duration}(y,t) = 0$ **do**

$t = t - 1$

$s_i.y = y$

$s_i.e = t$

$s_i.b = t - \Delta^{Duration}(y,t) + 1$

$i = i + 1$

$t = t - \Delta^{Duration}(y,t)$

$y = \Delta^{State}(y,t)$

end

Step 3. Finalization

Suppose that $s_1 \ldots s_q$ is the result of step 2, then the final best matched label sequence is given by $s_q \ldots s_1$.

end

Making use of semi-Markov conditional random fields, we present here a block diagram of our recognition system as in following figures.

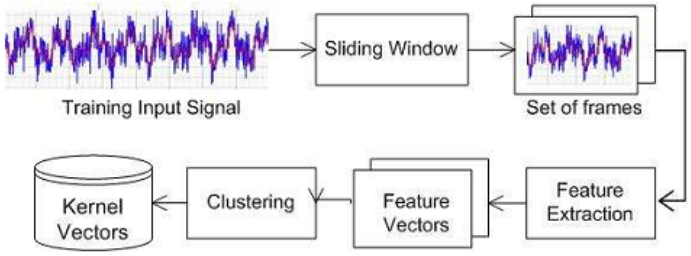


Figure 36 Quantization module

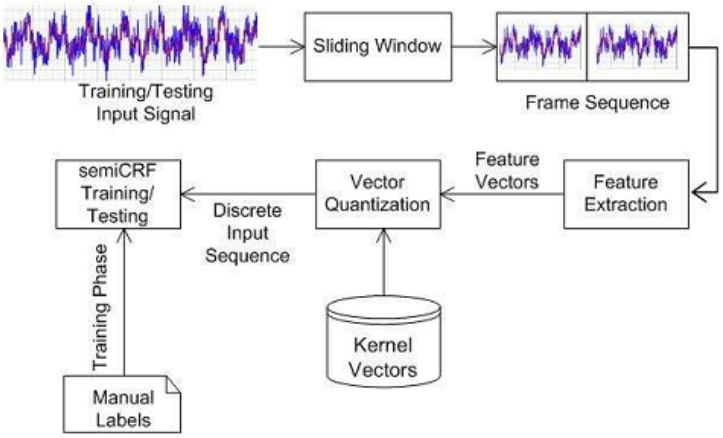


Figure 37 Recognition Module

The below figure illustrates our improvement in terms of complexity

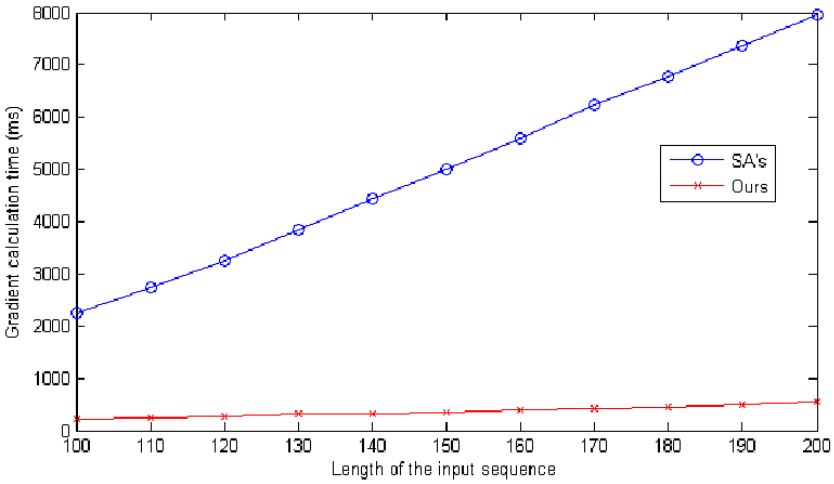


Figure 38 Execution Time

The blue represents excution time of the algorithm proposed in [CITE], our excution time is presented by the red. Obviously, our algorithm is remarkable faster than the previous one.

In our experiment, we use WiltiltV3 sensor to collect totally about 80 hours of daily activities including 4 expected routines: Dinner, Commuting, Lunch, Office work, the sampling rate is 20Hz, however after that we down the sampling rate to about 2.5 Hz in order to reduce the computation load. An example of our data collection tools on Pda is presented in the following picture.

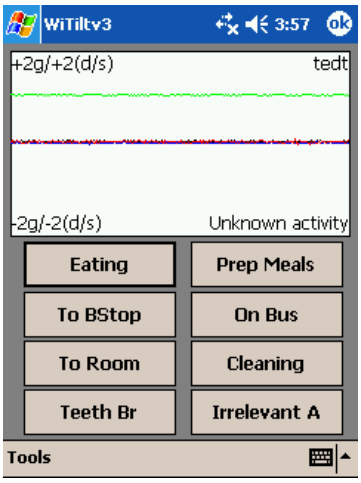


Figure 39 Data Collection Tool

Table 6, and Figure 40 demonstrate our recognition result.

Table 6 Recognition Result

Activity Name	Precision/Recall (%)
Dinner	76.60/63.91
Commuting	83.91/86.39
Lunch	89.79/97.24
Office	99.23/96.01

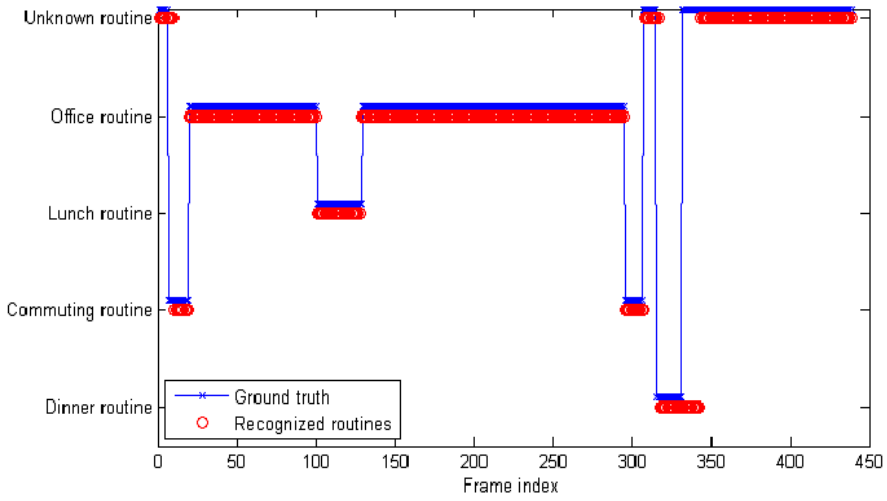


Figure 40 Recognition result

10 Ontology Engine

10.1 Introduction

Semantics is the meaning of a resource in the context in which it is referred while these meanings are provided by using modeling languages like rdf(s) and OWL. Modeling information in a formally structured and canonical representational format is called as ontology. So ontology is basically the representation of information combined together using some modeling language. Consider the example given below in Figure 41 (source: http://www.www2002.org/CDROM/refereed/506/). The semantics are provided by means of some modeling or representational scheme. Figure 1 shows that a *Person* can be a *Lecturer* or/and can be a *PhD Student*. For this modeling and representation ontologies are used.

Ontology provides two essential aspects that help to bring the web into its full potential. 1) Ontology defines a formal semantics for information allowing information to be process-able by computer system agents. 2) Ontology defines real-world semantics for resources, allowing them to link machine process-able content in a meaningful way based on consensual terminology.

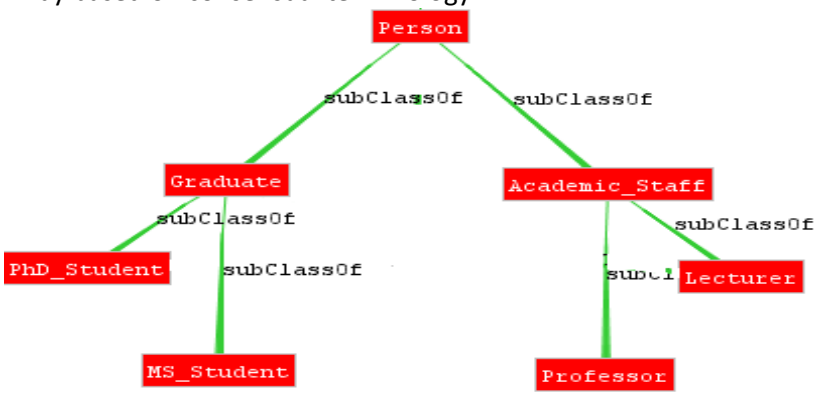


Figure 41 Ontology example

Ontology is formally defined as *an explicit and formal specification of a shared conceptualization* [Gru93, SiH05]. In this definition, *conceptualization* refers to an abstract model of some domain knowledge that identifies relevant concepts of the domain. *Shared* indicates that the knowledge captured in ontology is accepted by the community of that domain. The term *explicit* means that concepts and the constraints on these concepts are defined explicitly. Finally, *formal* reflects the notion

of using formally structured and canonical knowledge representation techniques in describing the body of knowledge. Now usage of ontologies is wide spread in Information Systems especially when building a *lingua franca* for resolving the terminological and conceptual incompatibilities between information networks of varying archetype and different provenance [Smi03]. One of the crucial tasks faced by practitioners and researchers in knowledge representation area is how to efficiently encode the human knowledge in ontologies? The proper maintenance of these, usually large, structured dynamic ontologies and in particular adaptation of these ontologies to new knowledge is a challenging problem in the Semantic Web research [70][72][74].

10.2 Use of Ontology in Activity Recognition

Ontology as discussed above is representation of information in such a format that is easily understandable and interpretable for computer system agents. Use of ontology provides basic description to events and activities, and from this information we can deduce new knowledge.

Use of ontology in activity recognition is relatively a new area of research. Using ontology help us better understand the activity in a given context. Activities recognized with the help of different sensors (i.e. body, location, motion, and video sensors) are low level activities and they are not in a capacity to be used for certain types of analysis and decision making. With the help of ontology, where we use the context information and link all the related activities in a chain, then with the help of customized rules we get the higher level activities that are more usable for decision making. Ontology helps in properly extracting the higher level activity of a set of activities in a series, e.g. series of low level activities like bending, sitting, jumping and walking with the help of ontology will result in a higher level activity i.e. exercising.

10.3 Proposed Solution

Ontology Engine is one of the main components of SC^3. SC^3-Ontology Engine (OE) is the process of inferring high level activities from low level activities recognized by different sensors. The component based framework architecture diagram of OE and the information flow is given in Figure 42, while the detail description of all the components are given in their corresponding sections. For instance, the Activity Recognizer component extract activity related information from XML and Text files, then with the help of context information available in Knowledgebase and customized rules we infer high level/actual activity performed by human body. Based on the activities performed, OE also gives suggestions and makes decisions in different environment with the help of context information available. For example; we have all the context/profile information (i.e. professor name, designation, current courses, class room no, and class timings) about a professor in the knowledgebase

(ontology). Now if Professor enters a class room on his class time, then the body, motion, location and video sensors will recognize that Professor has entered the class room at a specified time. Then the OE using these information from the sensors and information available in the Knowledgebase infers that its lecture time of Professor. So the system starts issuing commands for turning off class room lights, turn on computer and turn on plus scroll down multimedia (projector) in class room.

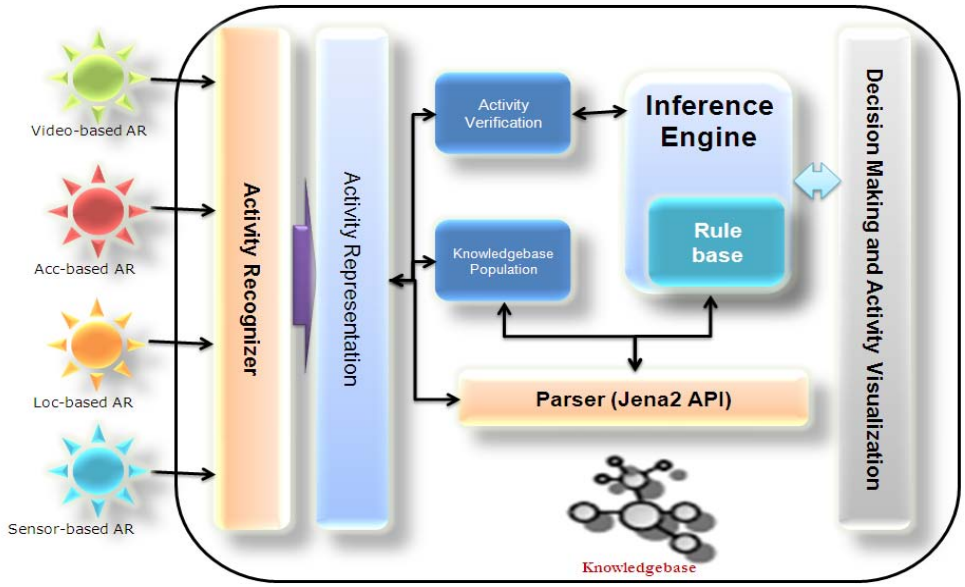


Figure 42 SC3-Ontology Engine; Inferring High Level Activities From Low Level Activities Using Context Information

❖ ***Activity Recognizer***

```
<?xml version="1.0" encoding="UTF-8"?>
<activities>
   <activity type="Motion">
              <detectedBy>Motion Sensor</detectedBy>
            <hasName>Prof. SY Lee</hasName>
              <activityName>Entering Class</activityName>
            <id>345</id>
            <time>2009:06:14:14:00:13</time>
   </activity>
   </activities>
```

Figure 43 XML output produced by motion sensor containing activity information

Activity Recognizer is a component that is responsible for recognizing data inputs from diverse sources. Video-based, sensor-based, motion-based, and location-based

activity recognition engines will provide output in different formats like XML (example given below) and simple text. So there is need of recognizer in the engine to properly parse all type of output produced by these different AR engines. Then it can extract the data from the file and provide these to the next module i.e. Activity Representation.

❖ ***Activity Representation***

As the extracted activity from the xml or text file will be stored in Knowledgebase and will also be used in Inference Engine to deduce higher level activity, so the activity needs to be formally represented in predefined semantic structure [ShE08]. For this reason, the Activity Representation component formally represent the activities that are recognized in the previous module, while the representation (see Figure 4) is provided by the Knowledgebase (explained later).

```
activityOnto:Activity_Instance_20090614140013345
a            activityOnto:Activity ;
activityOnto:hasConsequentAction    activityOnto:Action_Instance_145413546;
activityOnto:hasID                          345;
activityOnto:hasName                  "Entering Class";
activityOnto:hasType                          "Motion";
activityOnto:isA                        activityOnto:Room_Instance_Class;
activityOnto:performedAtTime            2009:06:14:14:00:13;
activityOnto:performedBy                      activityOnto:Person_Instance_345.
```

Figure 44 OWL representation (using N3 notation) of Activity (Person entering in a class)

The process of formal representation of activities is also important in a sense as we will have to check the activity for its consistency against the Knowledgebase and also it will be dump in the Knowledgebase for later use. Figure 44 is the formal representation of the activity given in the xml representation above in Figure 43, here the representational schema is provided by the Knowledgebase.

❖ ***Knowledgebase***

Knowledgebase (KB) serves as the back bone of OE. It is responsible for proper communication of information among all the components of OE. It stores all the possible types of activities that a human body can perform in different context/situation, with the information of different activities priority for different users and group of users.

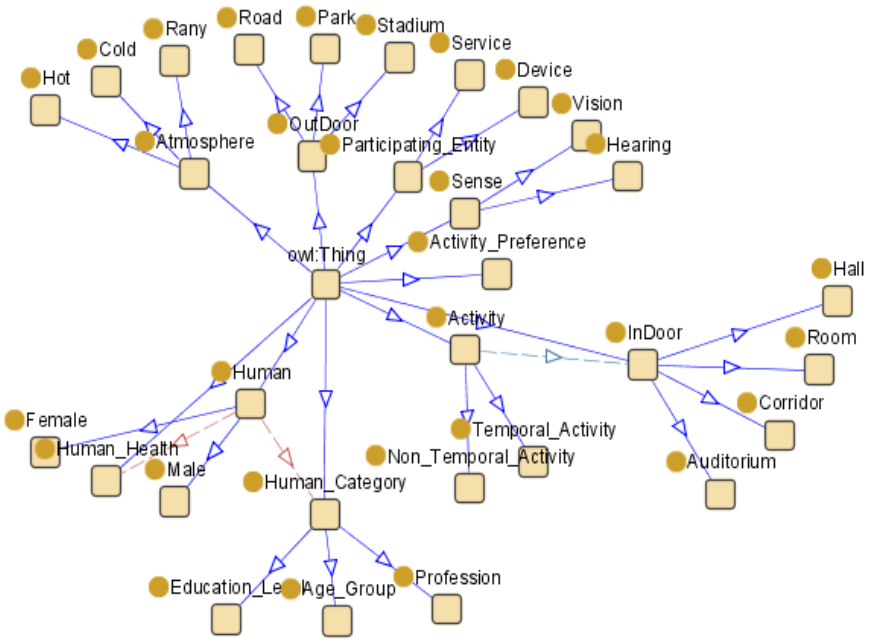


Figure 45 Knowledgebase (Human Activities ontology)

The proper engineering of the KB is most important activity in the development of OE. To engineer the KB (see Figure 45) we have to look at the same problem from different directions, for example; if a building has caught fire then no one is allowed to enter that building, but firemen are allowed to perform their work and even enter the building. So we need to introduce some type of priority for different actions that a particular group of human can perform in a given context while the other cannot.

When an activity is recognized by the sensors then this knowledgebase is parsed and if required in different situations then inferencing is done for decisions against activities. So it is actually the ontology, where all the activities are semantically modeled and available for analysis and decision making.

❖ ***Activity Verification***

Activity Verifying is important for two reasons:

Check for the consistency of the newly recognized activity against the knowledgebase developed for the activities. In consistency verification, the activity represented in Activity Representation component is verified against the KB for its structure and the information contained in it. So basically the verification is made to check that weather the activity fulfill all the requirements or not. So if consistency verification is positive the other modules work on that activity.
Secondly after consistency verification, the existence verification is done for the

activity that is this activity already present in the knowledgebase or not? If not present then it is given to Knowledgebase Population module to store this in the Knowledgebase.

❖ ***Knowledgebase Population***

Knowledgebase (ontology) evolution is of two types i.e. ontology enrichments and ontology population [FPG06, CFH06].

Ontology Population: When we get new instances of concept(s) already present in the ontology. Then this concept(s) is not inserted for the second time. Here only the new instance(s) of this concept(s) is introduced and the ontology is populated.
Ontology Enrichment: When we get new concept(s), which is totally new for our ontology or it does have some sort of changes from its counter concept(s) in the ontology. Then we enrich our ontology to accommodate the new changes and also populate our ontology for its instance.

Here our focus is on ontology population where new instances are introduced in the Knowledgebase against the already present concepts. Knowledgebase Population module is responsible to store all the newly recognized activities in the KB for later use, where this logging of activities in the Knowledgebase is achieved with the help of Parser.

❖ ***Parser***
For any type of information manipulation from the Knowledgebase, Parser is responsible to properly handle all the operation regarding that matter. The Parser normally communicates with Activity Representation component to properly represent the activity, it also parse the Knowledgebase for the Inference Engine for verity of different reasons like verification of activity and decision making, To populate the KB for newly recognized activity, the Parser is also used in that case.

❖ ***Inference Engine***

To understanding the context of an activity and to extract high level (abstract) activities from low level activities recognized by sensors, we need to have an Inference Engine for analysis of these activities and to make proper decisions on behalf of human users. So the Inference Engine is very important component of OE. It uses the activities information with respect to their context information and infers high level activities. The decisions or suggestions of Inference Engine are very much dependent on the domain and user intensions. So for this reason we also introduced the user defined customized rules in the inferenceing process. E.g., if a person is falling from a building in a stadium then its context maybe that there is a jumping competition, but if a person if falling from a building and that building is of some educational institute then its context is that there is an emergency situation over there. So we need to have domain specific customized rules.

❖ ***Rule base***

Every organization have their own customized rules; e.g., in Kyung Hee University (KHU), Korea, one course can only once be studied in a program. Now for example if a person *Tea Ho* is taking the Data Mining course for the second time and he has got B+ or above grade previously, so according to KHU rules it is not possible to take this course again in the same program.

So for these sorts of situations and actions, we need to define customized rules for different activities.

❖ ***Decision Making and Activity Visualization***

After the process of inferenceing, the system can take decisions or give suggestion against different activities. So this module is responsible for performing some actions against the suggestions made by the Inference Engine. This module also visualizes the activities and the Knowledgebase for proper understanding of the activities.

10.4 Implementation and Results

In this section we discuss the implementation details of the system (Ontology Engine). Till first phase demonstration of the SC^3 we have implemented the system for 14 different activities and with 11 customized rules. We did implement all the components of OE comprehensively and they are easy to extend. The number of activities for OE is dependent on the number of activities recognized by the different sensors.

We have developed the OE using Java with Netbeans-6.0-IDE, and for manipulation of ontology we used Jena2, Protégé, and ProtégéOWL API's. The inferenceing part is implemented with the help from inference engine Pellet 3.4.

To extract the activity information from the text document we used *File* and *FileInputStream* classes available in java, while in case of xml files, we used DOM from w3c. Then after that we used Jena2 for parsing the triple store (Knowledgebase). We also used Arq API to query the Knowledgebase for different activities. The Knowledgebase Population is achieved with the help of both Jena2 and ProtegeOWL API's. The same way before the population of Knowledgebase, we made the existence and consistency verification of the activities (see Figure 46 for code of existence verification of an activity).

```
boolean flag=false;
owlDomMdl= ProtegeOWL.createJenaOWLModelFromURI(MHBBAD.uri.toString());
for(Iterator it=owlDomMdl.getOWLNamedClass("Activity").getInstances(true).iterator();it.hasNext();)
{
    OWLIndividual objInd=(OWLIndividual) it.next();
    if(objInd.getName().equals("Activity_Instance_" + time + id))
    {
        flag=true;
        break;
    }
}
```

Figure 46 Code for checking existing verification of an activity

```
"SELECT ?activityName ?hasConsequentAction ?type ?performedBy ?performerName ?time ?actionDes
?performedAt ?performedAtLoc ?hasType ?actionTime WHERE { <" + strNS + strActivity + "> <" + strNS +
"hasName>    ?activityName ." +
"<" + strNS + strActivity + "> <" + strNS + "hasConsequentAction> ?hasConsequentAction ." +
  "<" + strNS + strActivity + "> <" + strNS + "hasType> ?type ." +
  "<" + strNS + strActivity + "> <" + strNS + "performedAtTime> ?time ." +
"OPTIONAL {<" + strNS + strActivity + "> <" + strNS + "performedBy> ?performedBy} ." +
  "OPTIONAL {?performedBy <" + strNS + "hasName> ?performerName} ." +
  "?hasConsequentAction <" + strNS +    "hasActionDescription>    ?actionDes ." +
"?hasConsequentAction <" + strNS + "hasType> ?hasType ." +
"?hasConsequentAction <" + strNS + "hasTime> ?actionTime ." +
"OPTIONAL {?hasConsequentAction <" + strNS + "hasPerformedAt> ?performedAt} ." +
"OPTIONAL {?performedAt <" + strNS + "hasName> ?performedAtLoc}}";
```

Figure 47 SPARQL query to extract all the corresponding information of an activity

To get information about some specific activity and their consequent actions, we wrote SPARQL queries that are executed using Jena2 API while using the functionality of Arq API. Figure 47is a query of getting the information for some particular activity and their consequent action. For this query, the activity is provided by the system or user and then its corresponding information are all extracted.

To implement the Inference Engine, we have used the inferenceing at A-Box [69] level (i.e. at instance level). This technique uses the instances and makes the decisions and suggestions based on instance matching. As discussed above, we have implemented 11 different customized rules for 14 different activities. For rule please see Table 1. Now based on these rules we made all the decisions and for all the 14 activities we got correct results. For the implementation of these rules, we used the forward chaining method i.e. one activity having dependency on the next one and so on.

For decision making based on the customized rules, we also achieved results like turning on the lights and TV when it was required. We have also generated the alarm for user based on his current activity and future activity. For better understanding of activities and finding out their relationships with one another, we also have provided the facility of visualization. To visualize the Knowledgebase, we have extended the *TouchGraph* API for graph drawing in order to visualize the graph view

of the ontology structure. Resources, such as classes, are depicted as nodes, where these connected through properties which are depicted as the edges in the Visualization. A modified version of the *Spring* graph drawing algorithm is implemented here for visualization that ensures esthetically good looking graph see Figure 48.

Table 7 Customized Rules

Rule1 Activity(a1) ⊓ hasContents(eating) ⊓ hasNextActivity(null) → Activity.Create(a1)
Rule2 Activity(a1) ⊓ hasContents(taking medicine) ⊓ hasNextActivity(null) → Activity.Create(a1)
Rule3 Activity(a1) ⊓ hasContents(reading) ⊓ hasNextActivity(null) → Activity.Create(a1)
Rule4 Activity(a1) ⊓ ¬ hasContents(taking medicine) ⊓ hasNextActivity(a2) ⊓ Activity(a2) ⊓ hasContents(eating) → Activity.Create(a1) ⊓ Activity.Create(a2) ⊓ reminder(take medicine)
Rule5 Activity(a1) ⊓ hasContents(reading) ⊓ hasNextActivity(a2) ⊓ Activity(a2) ⊓ hasContents(TV On) → Activity.Create(a1) ⊓ Activity.Create(a2) ⊓ turnOff(TV)
Rule6 Activity(a1) ⊓ hasContents(doing exercise)⊓ hasNextActivity(null) → Activity.Create(a1) ⊓ turnOn(music)
Rule7 Activity(a1) ⊓ hasContents(unknown exercise) ⊓ hasNextActivity(null) → Activity.Create(a1) ⊓ reminder(movements are wrong)
Rule8 Activity(a1) ⊓ hasContents(entering kitchen) ⊔ Activity(a2) ⊓ hasContents(entering bedroom) → Activity.Create(a1) ⊔ Activity.Create(a2) ⊓ turnOn(lights)
Rule9 Activity(a1) ⊓ hasContents(leaving kitchen) ⊔ Activity(a2) ⊓ hasContents(leaving bedroom) → Activity.Create(a1) ⊔ Activity.Create(a2) ⊓ turnOff(lights) ⊓ turnOff(TV)
Rule10 Activity(a1) ⊓ hasContents(in the kitchen)⊓ hasNextActivity(null) → Activity.Create(a1) ⊓ turnOn(TV)
Rule11 Activity(a1) ⊓ hasContents(sit down) ⊓ hasNextActivity(a2) ⊓ Activity(a2) ⊓ hasContents(looking at TV) → Activity.Create(a1) ⊓ Activity.Create(a2) ⊓ turnOn(TV)

10.5 Limitations

As we have demonstrated the 1st phase results of SC3, so there are some limitations in our system. Here we will discuss the limitation of Ontology Engine which still needs to be address to get more covered and concrete results.

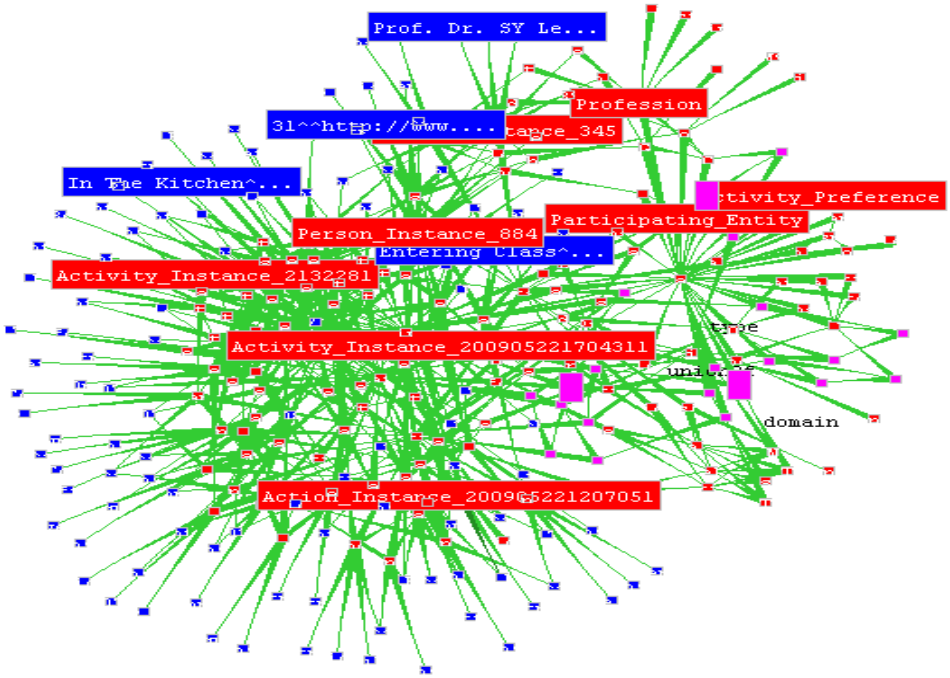


Figure 48 Ontology Engine Output (Activity Visualization)

As discussed earlier, the numbers of activities recognized by different sensors are limited in number and these needs to be increased to get the more useful high level activities.

Currently, we are working on A-Box inferenceing, which is more time consuming where for life care system we need to have good response time. So we are planning to use A-Box in integration with T-Box.

The rules are also limited in number, it also need to be increased.

In 1st phase of the system, we used the forward chaining method for inferenceing higher level activities, while we can also get some useful information if we apply the backward chaining. But proper management for backward chining is a major issue. We are working on this issue to reach to some feasible tradeoff in this situation.

One of the main problems is the result accuracy of the activity recognition engines. If they keep on producing wrong results then the system performance keeps on decreasing.

Missing information is also another problem [78], sometimes we don't have the context information for specific activity then in that case the system results are unpredictable.

Chapter 11

IMPLEMENTATION

11 Implementation

11.1 System Workflow

In the first implementation stage, we have developed the following main components as shown in Figure 49: Cloud gateway to deliver raw data from WSNs, a TCP/IP communication module via Socket to transmit data, a Context-aware Engine (including Sensor-based AR, Video-based AR, Context-Awareness, Ontology Engine), a Service Manager, Security Manager (Authentication and Access Control), an Web2.0 interface for nurses and doctors to access patient medical information on the Cloud.

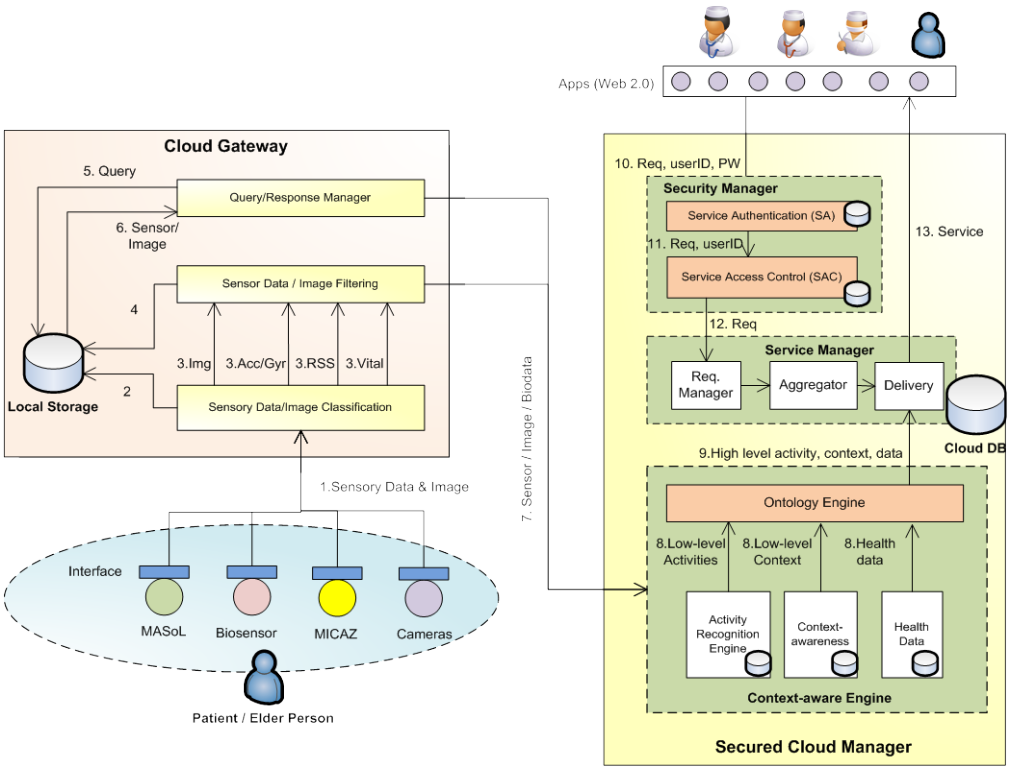


Figure 49 Functional Architecture of SC[3]

The system work flow as depicted in Figure 49 as follows:

1) Sensed data and video are captured from an embodied sensor and cameras, then being transmitted to the home Cloud gateway.
2) The Gateway classifies each type of data and store locally.

3) Classified data is forwarded to the Filtering Module, where it is filtered unnecessary data before forwarding to the Cloud to increase the communication performance.
4) Filtered data is also updated to the local database for later use.
5) If there is a query from the services/applications, the Query/Response Manager queries data from the local database
6) and responses to the requesters.
7) Data is transmitted to the Cloud via TCP/IP socket.
8) In the Cloud, raw data is used to deduce user activity and context, such as the patient is walking, eating, staying in the kitchen
9) Activity and context are forwarded to ontology to represent in easy manner, and deduce higher level activity and context. The ontology also makes decision to response to some context, for example if the patient is reading the book, then TV should be turned off.
10) When doctors, nurses access data, they must authenticate.
11) After successful authentication, the Access Control module makes decision whether his/her access permission is allowed or not.
12) If yes, it allows him/her to access to the Cloud data.
13) Data is forwarded to authentic nurses and doctors.

11.2 UML Class Diagram

This section presents class diagrams of each component in SC^3.

❖ ***Overall Sequence Diagram***

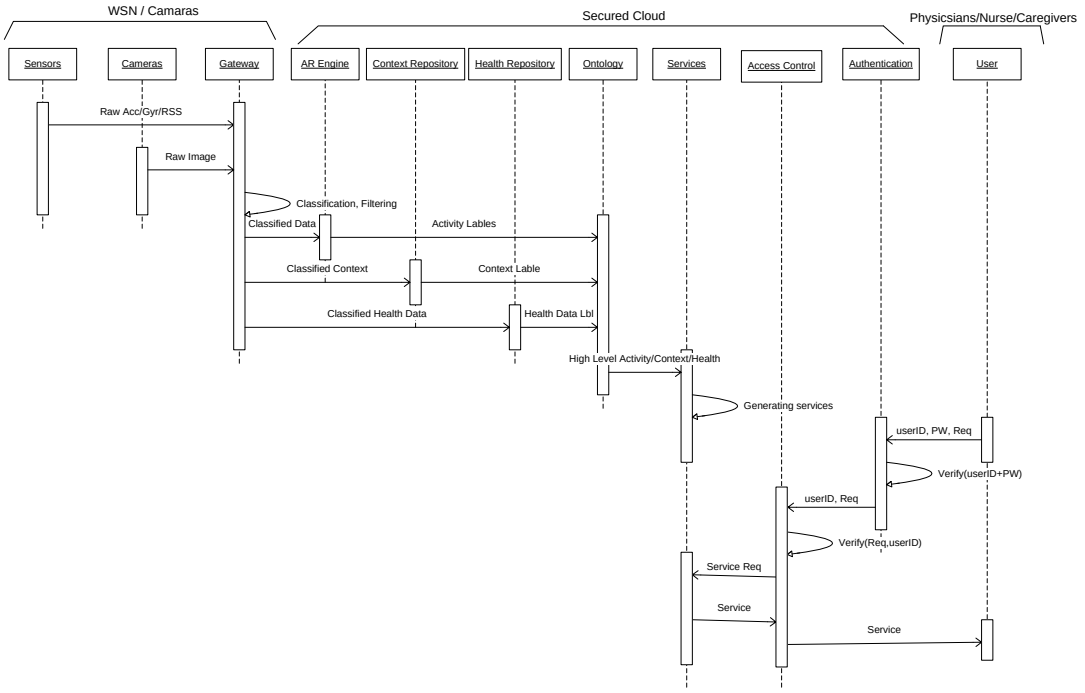


Figure 50 Overall UML Design

❖ *Cloud Gateway*

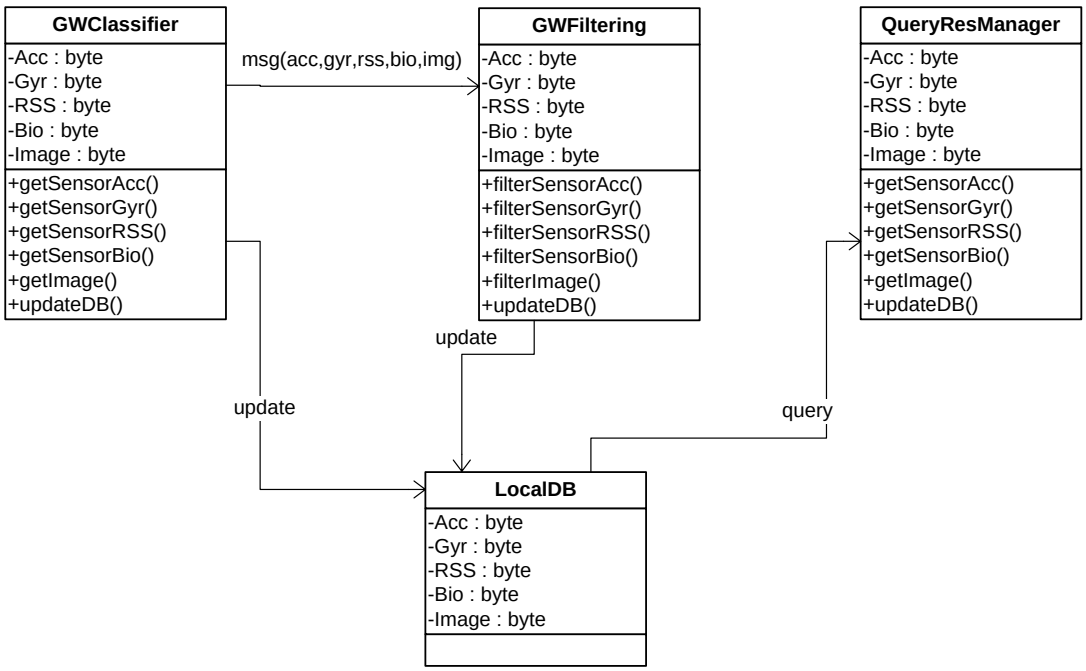


Figure 51 UML Design of Cloud Gateway

❖ ***Sensor-based Activity Recognition***

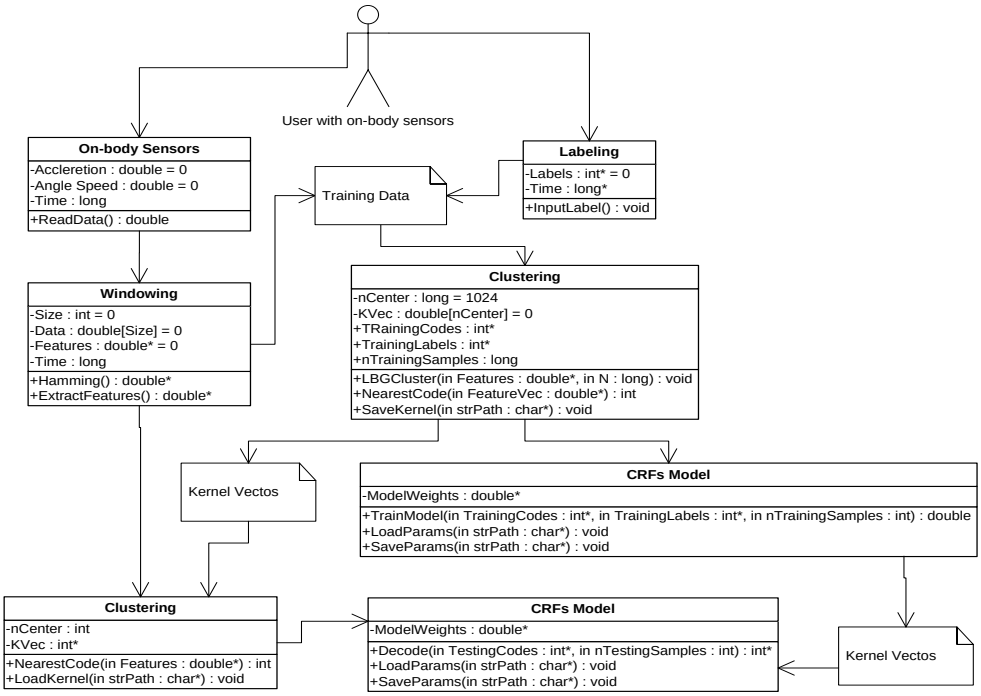


Figure 52 UML Design of Sensor-based Activity Recognition Engine

❖ ***Video-based Activity Recognition***

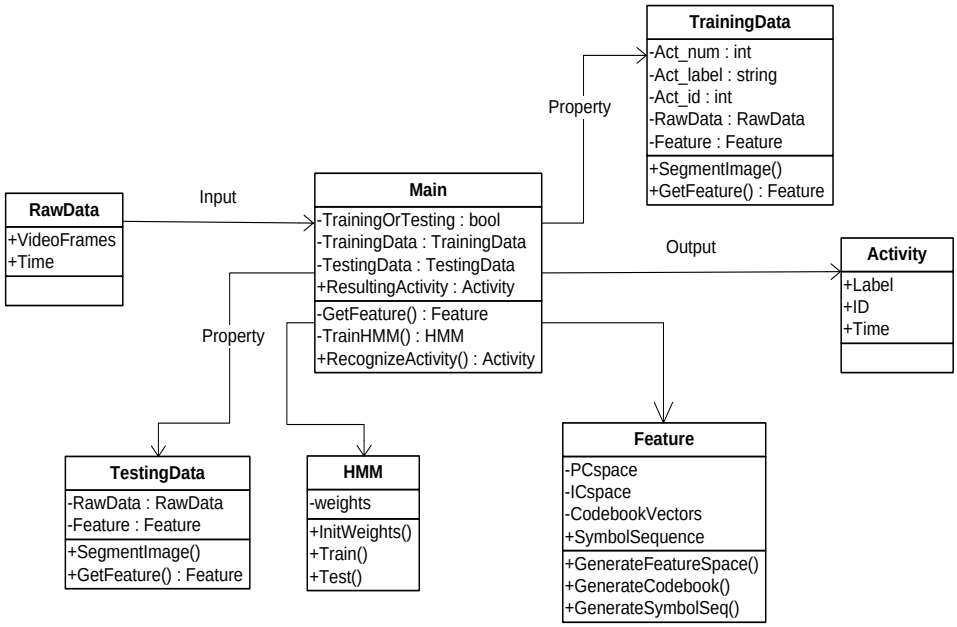


Figure 53 UML Design of Video-based Activity Recognition

❖ ***Ontology Engine***

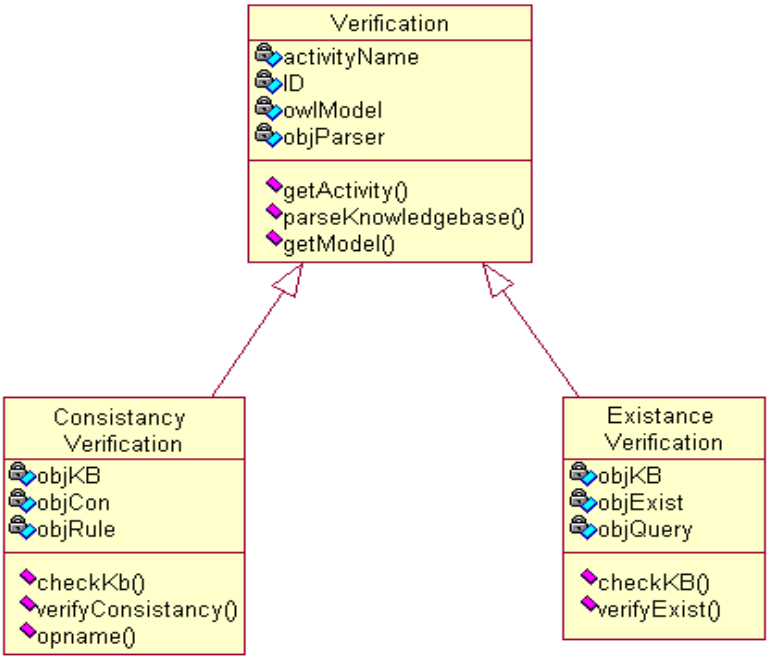


Figure 54 UML of Activity Verification Module

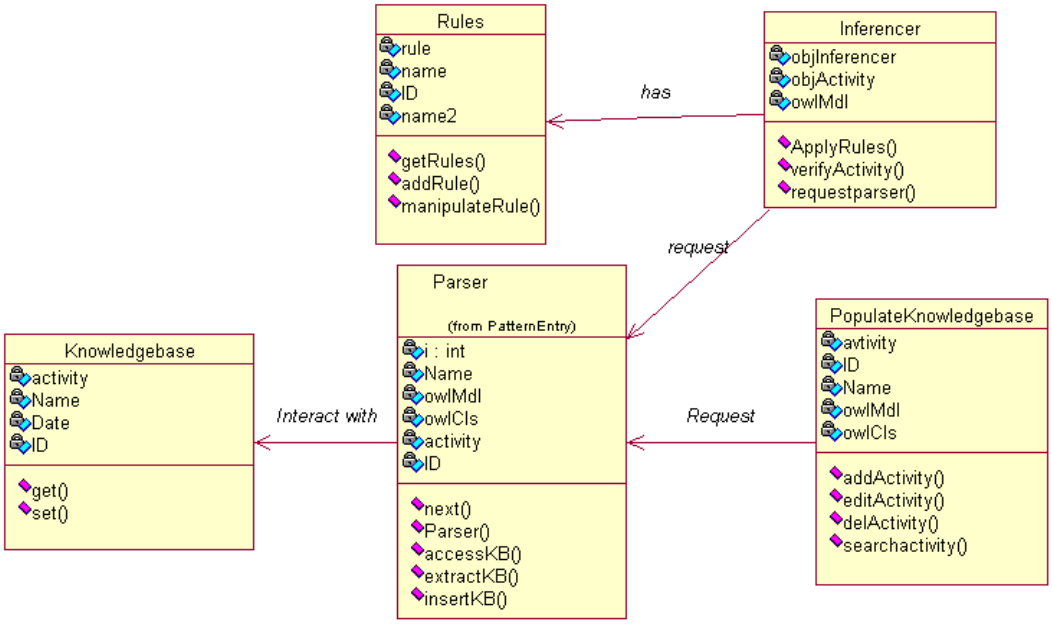


Figure 55 UML of Knowledgebase Manipulation

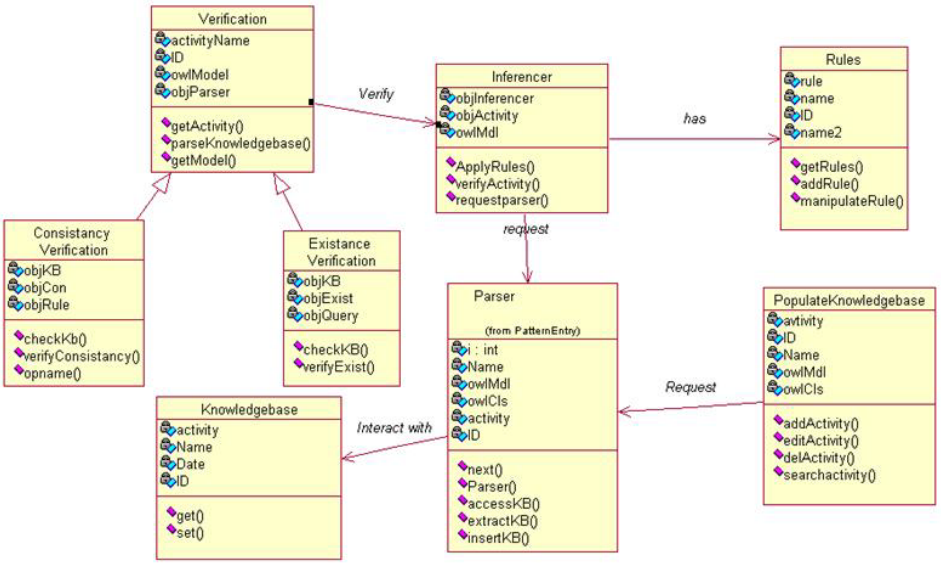


Figure 56 UML Design of Activity Verification and Manipulation Module

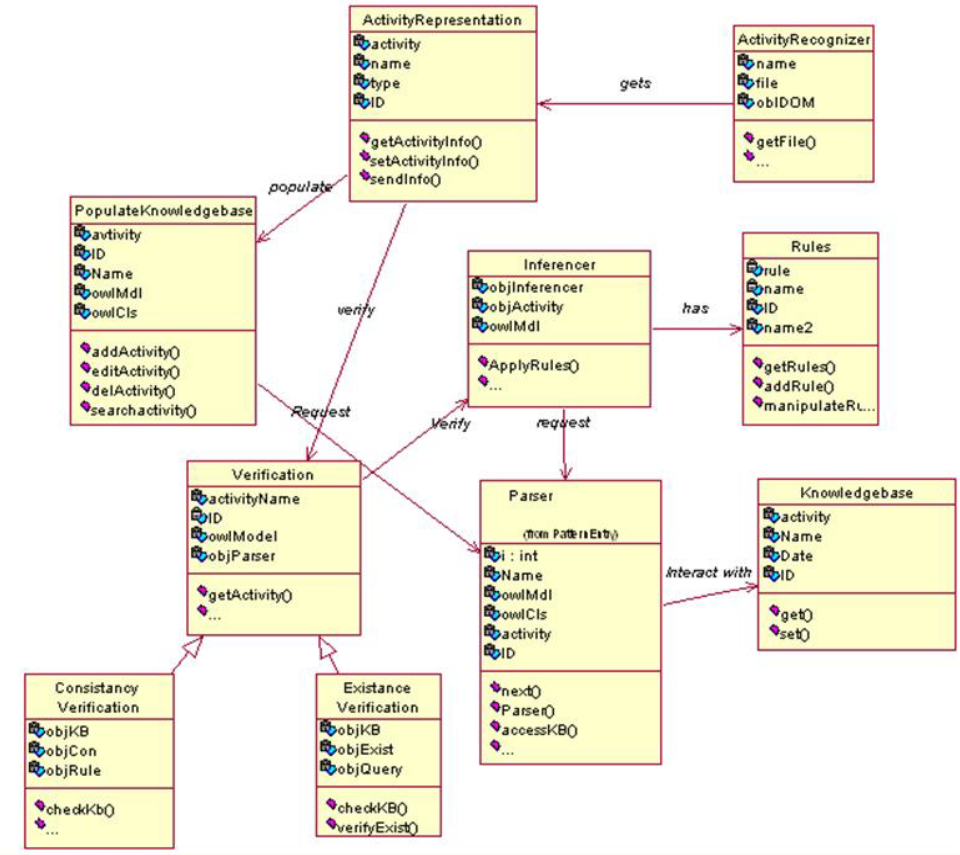


Figure 57 UML Design of Ontology Engine

❖ ***Activity-based Access Control***

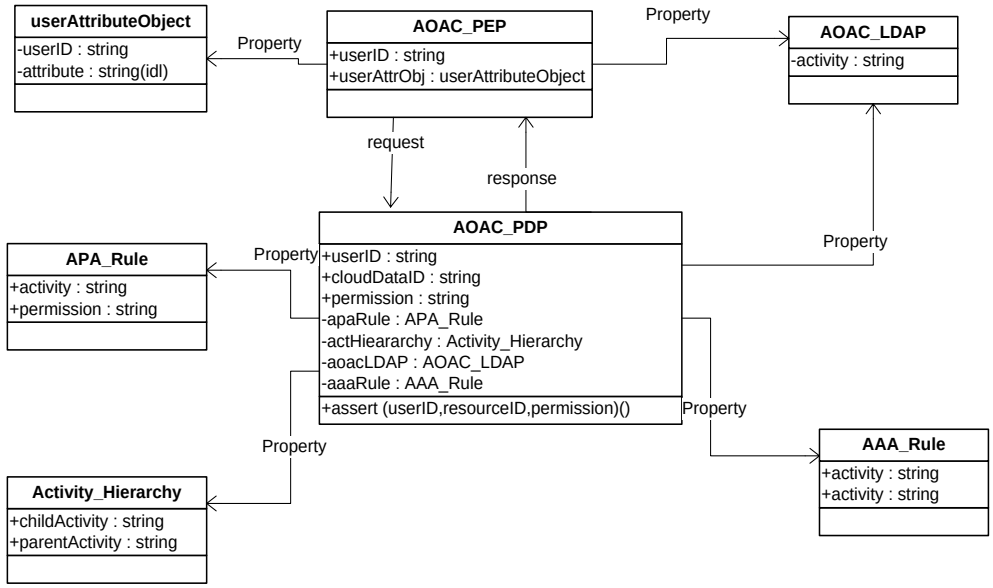


Figure 58 UML Design of Access Control Module

11.3 UML Sequence Diagram

❖ ***Cloud Gateway***

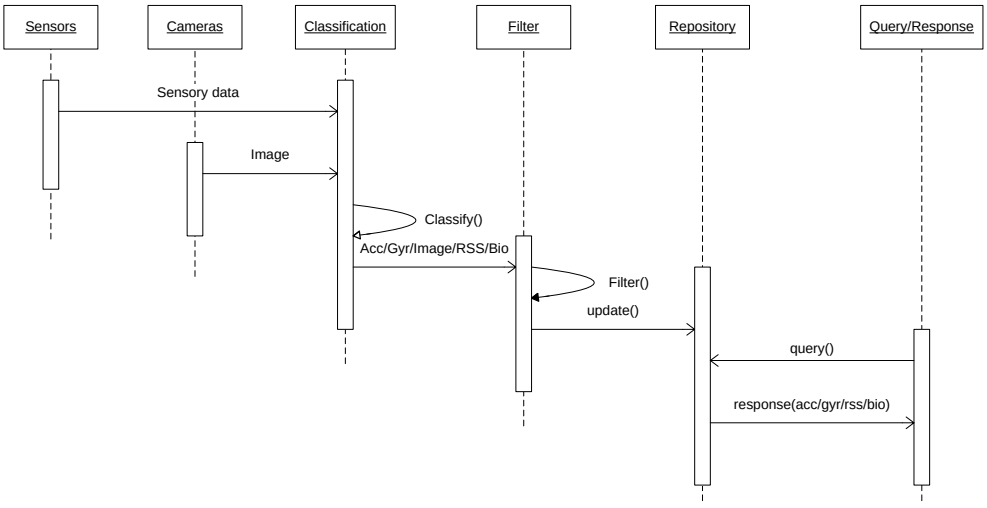


Figure 59 Sequence Diagram of Cloud Gateway

❖ ***Video-based Activity Recognition***

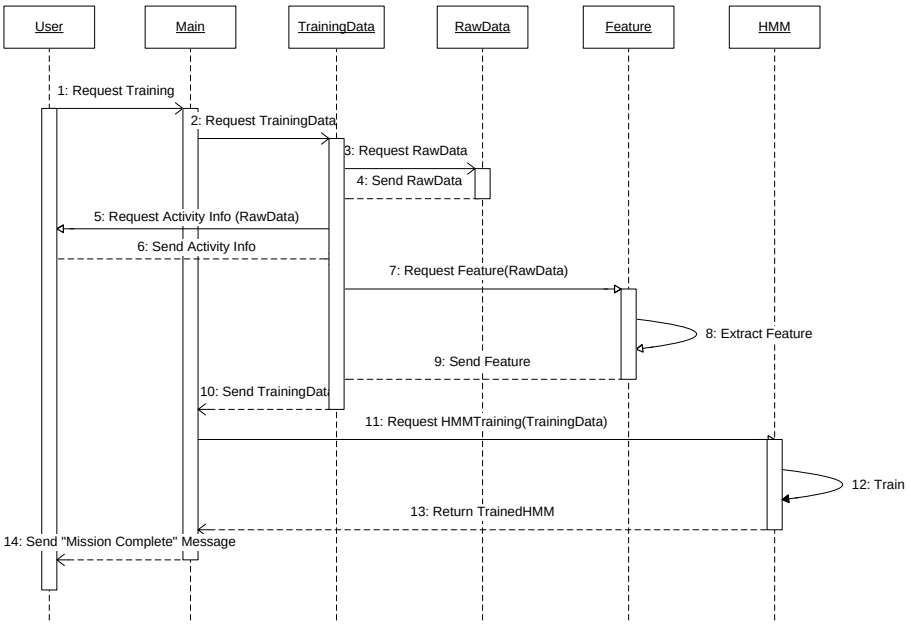


Figure 60 Sequence Diagram of Training Phase

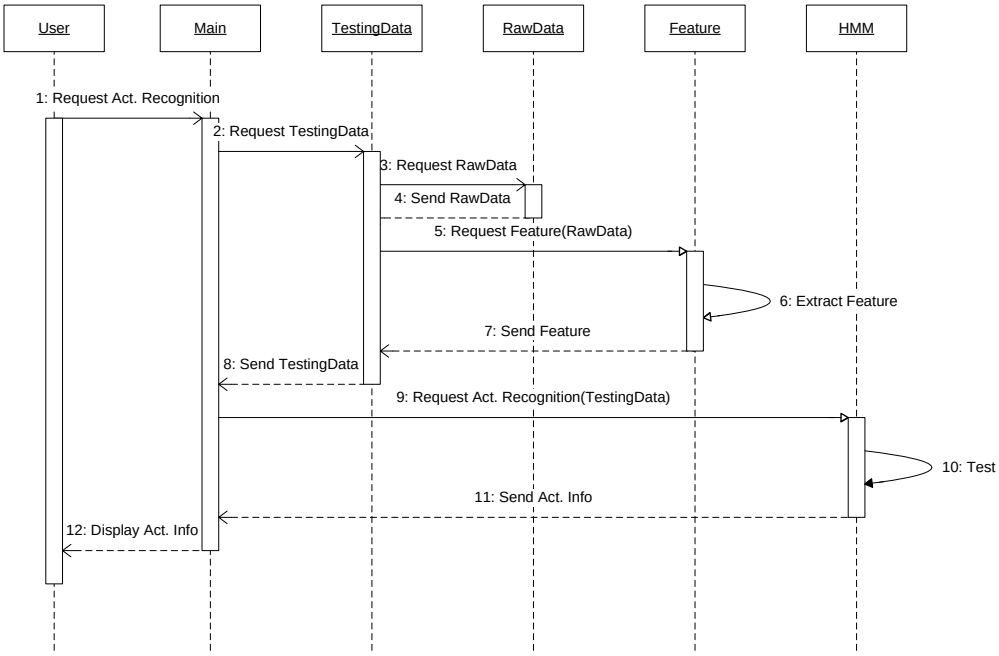


Figure 61 Sequence Diagram of

11.4 Scenario Design – SC^3 Supports Alzheimer's Disease

Our general system deployment is shown in Figure 63. The patient's house includes a kitchen, a bed-room, and a living room. Several sensors and cameras are deployed in the patient's house to collect sensory data and images. We deploy a cloud gateway in the living room to collect data from all sensors and cameras. It connects to the Cloud via Internet high speed router. Doctors, nurses, and patient's relatives (e.g. his daughter) can access easily via Web2.0 interface.

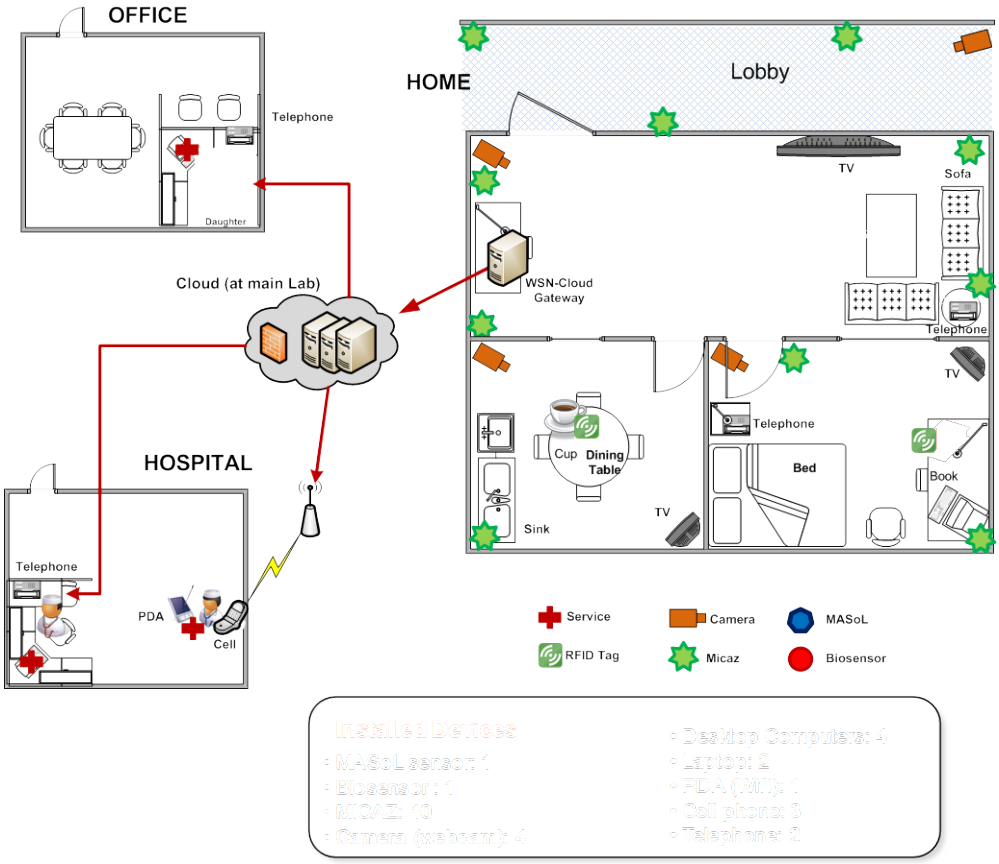


Figure 62 Overall Scenario Design at ITRC

The following figures show how we collect data from sensors and cameras and deploy to the Cloud.

❖ ***Video-based AR***

- **Output**
  - Real-time walking, sitting, lying.
- **Technology**
  - Camera wire transmission
  - 2D image processing
- **Devices**
  - High quality camera (4)
- **Approach**
  - Uses Video frame to infer user's motions
  - WCG sends user's motions to Clouds

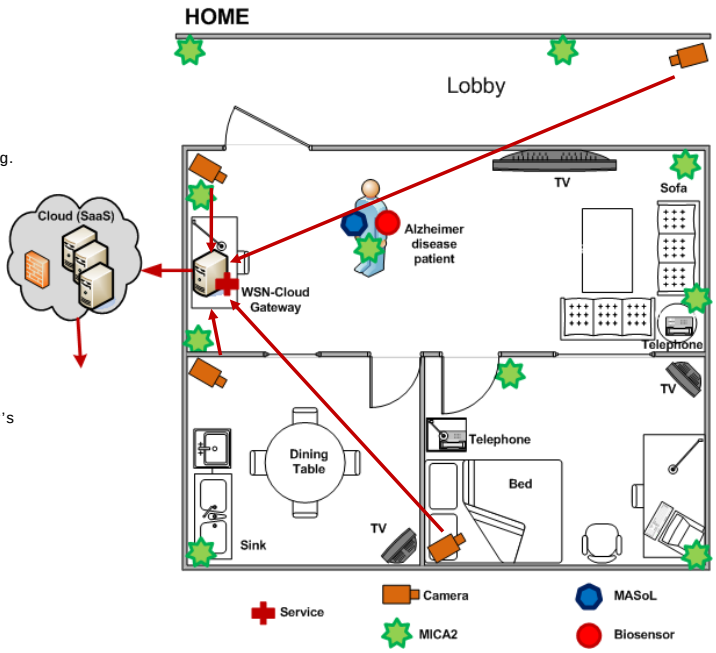


Figure 63 Camera Deployment for Video-based AR

❖ ***Sensor-based AR***

- **Output**
  - Off-line activities (eating, taking medicine)
- **Technology**
  - Gyroscope, Accelerometer
- **Devices**
  - MASoL (1 / person)
- **Approach**
  - Collecting gyroscope and accelerometer information
  - Upload to Clouds
  - At the end of the day, infer what user was doing

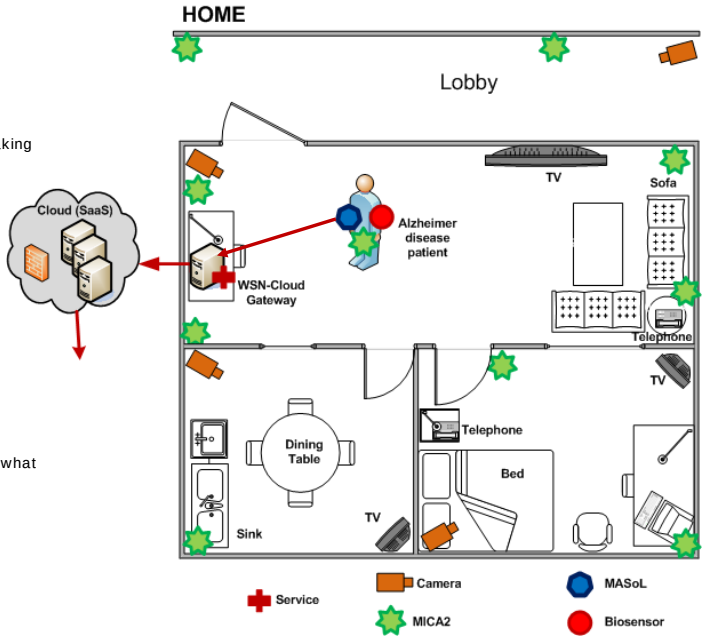


Figure 64 Sensor Deployment for Sensor-based AR

❖ *Location Tracking*

- **Output**
  - Current user location
- **Technology**
  - Zigbee
  - RSS (Radio Signal Strength)
  - RSS Localization Algorithm (RLA)
- **Devices**
  - MASoL (1 / person)
- **Approach**
  - User carries a MASoL device
  - Using RSS to measure distance from surrounding sensors to the sensor attached on body
  - RLS algorithm infers user's location based on different distances
  - WSN-Cloud Gateway (WCG) sends location to the Clouds

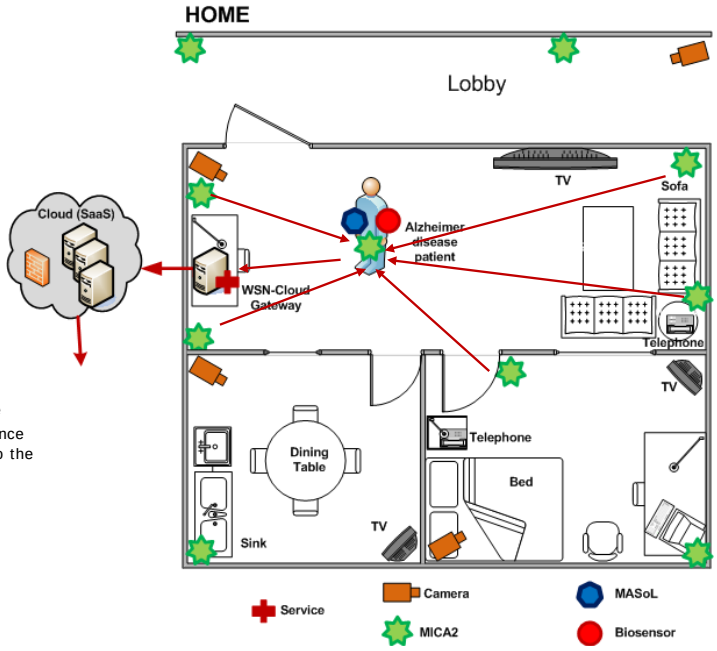


Figure 65 Location Tracking Deployment

❖ *Medical Data Collection*

- **Output**
  - Medical status (heat beat, blood pressure, body temp)
- **Technology**
  - Zigbee
  - Medical monitoring
- **Devices**
  - Medical sensor (1 / person)
- **Approach**
  - User medical data is collected to WCG
  - WCG sends to Cloud

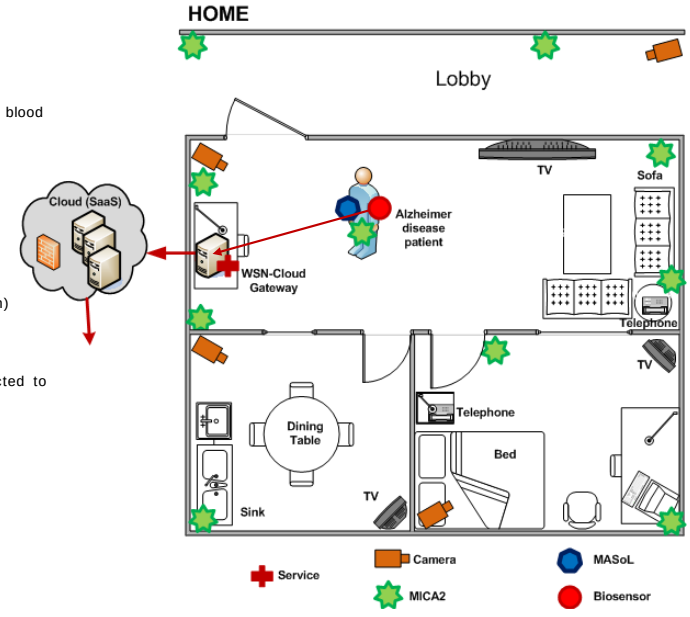


Figure 66 Biosensor Deployment for Medical Data Collection

❖ ***Authentication and Access Control to Cloud Data***

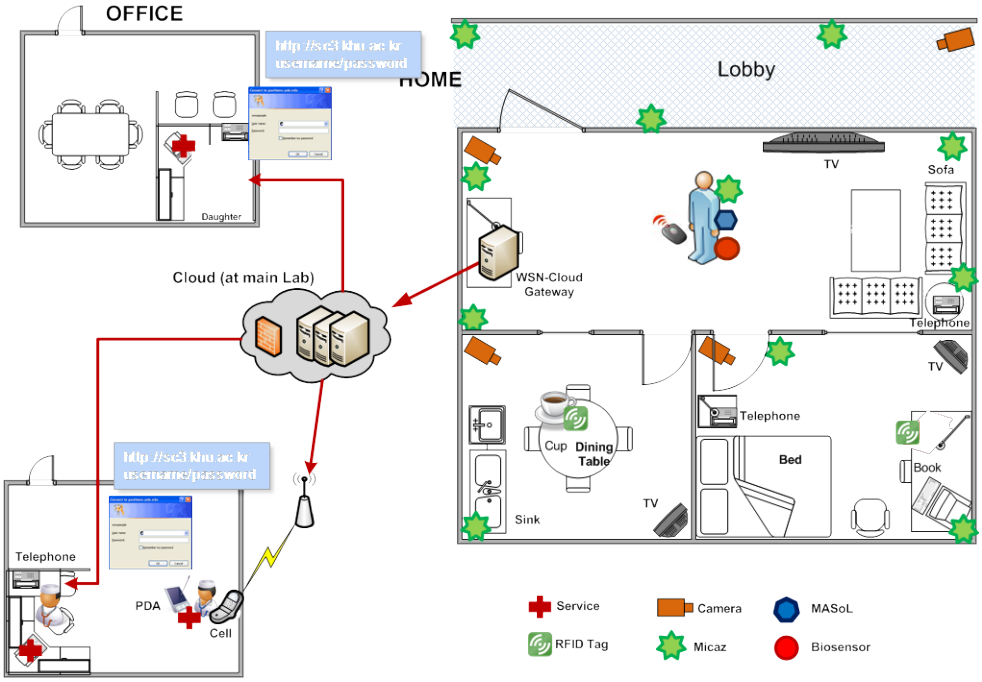
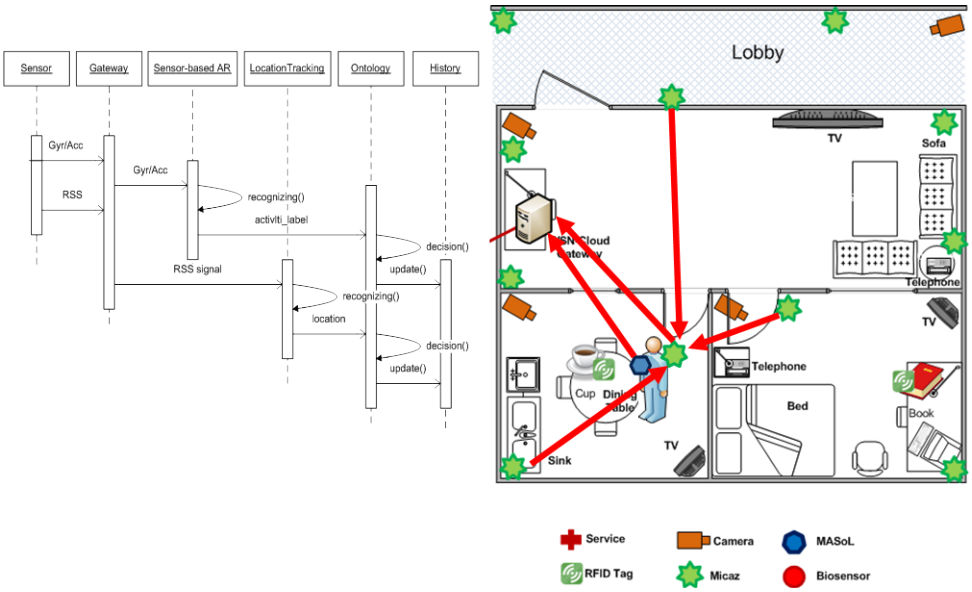
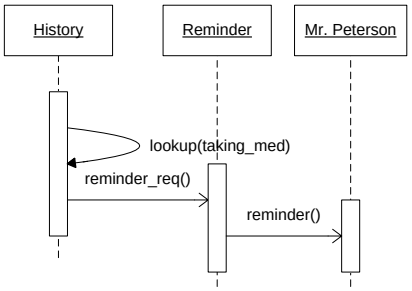


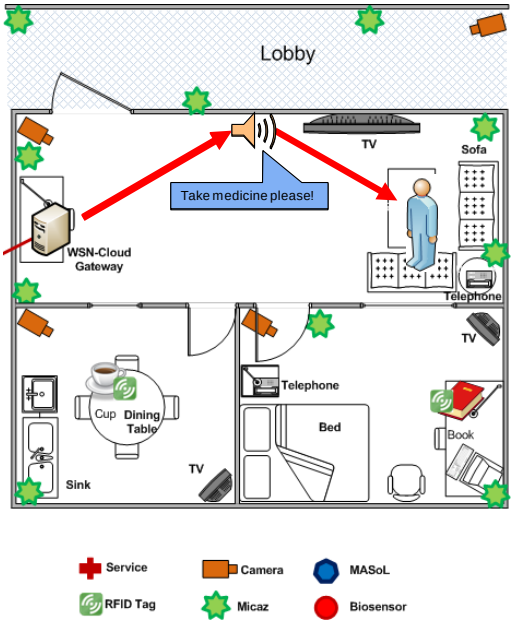
Figure 67 Authentication and Access Control to Cloud Data

11.5 Scenario Flow

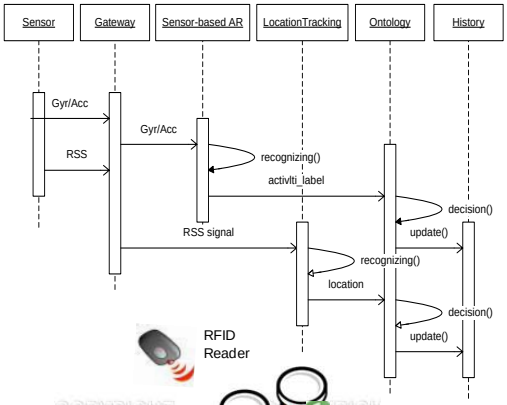


8:00AM: Reminder for taking medicine

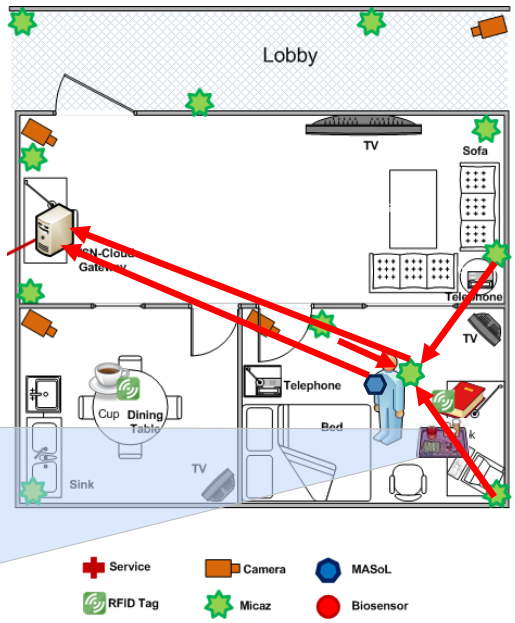




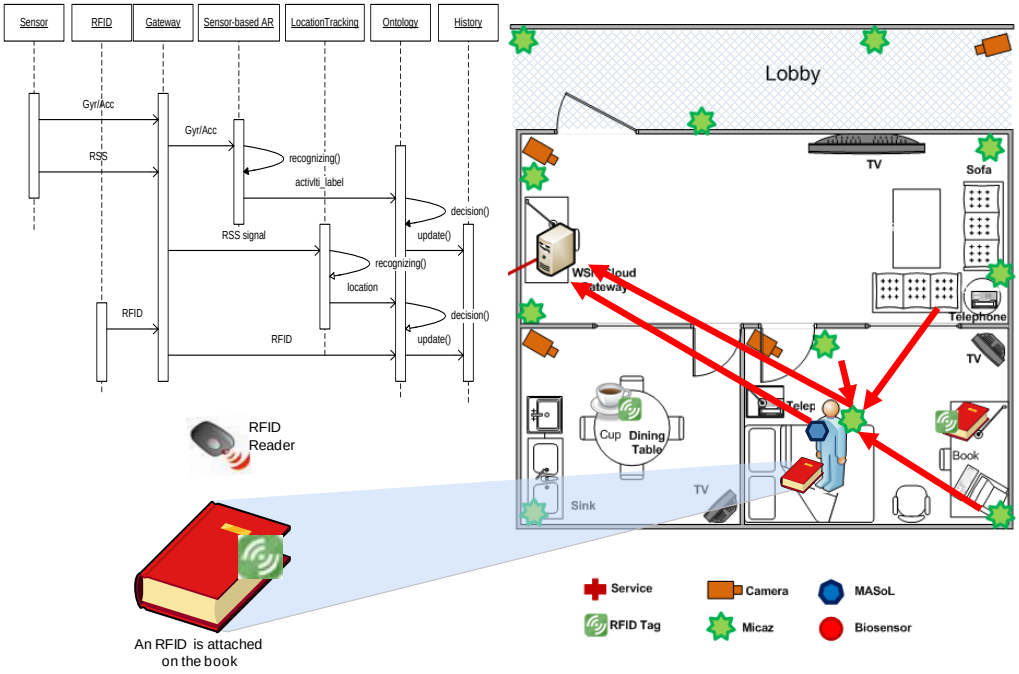
8:15AM: Taking medicine



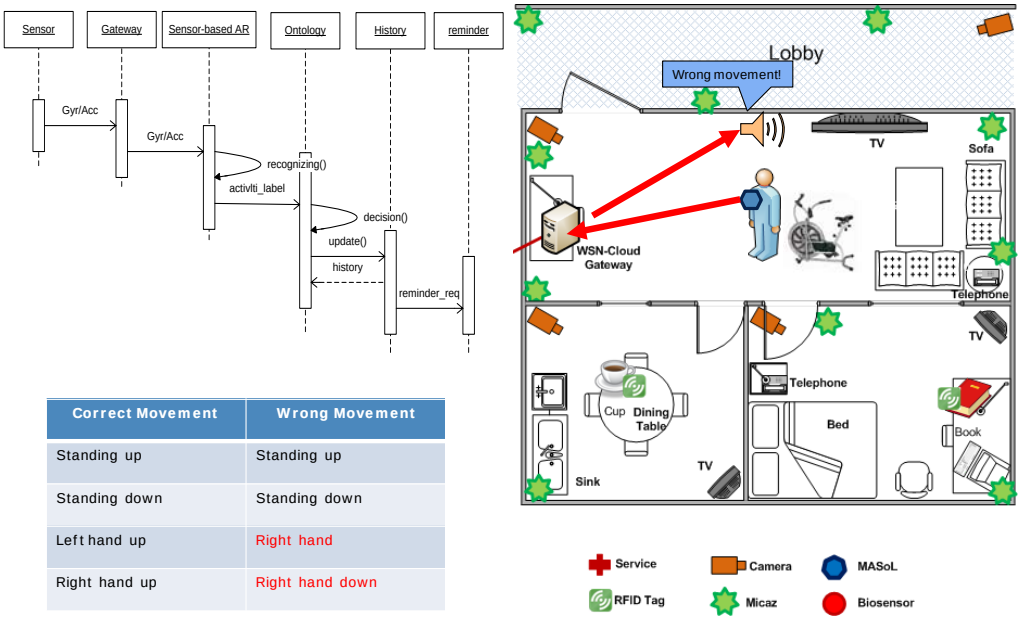
Medicine Tray: each medicine type is attached with an RFID



8:30AM: Watching TV & Reading a book

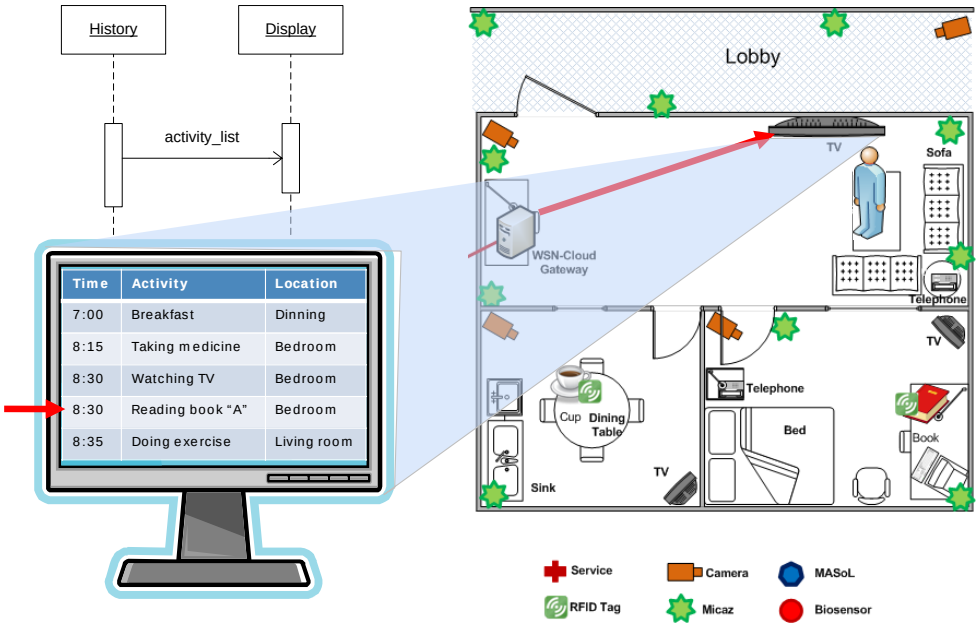


9:00 AM: Doing exercise

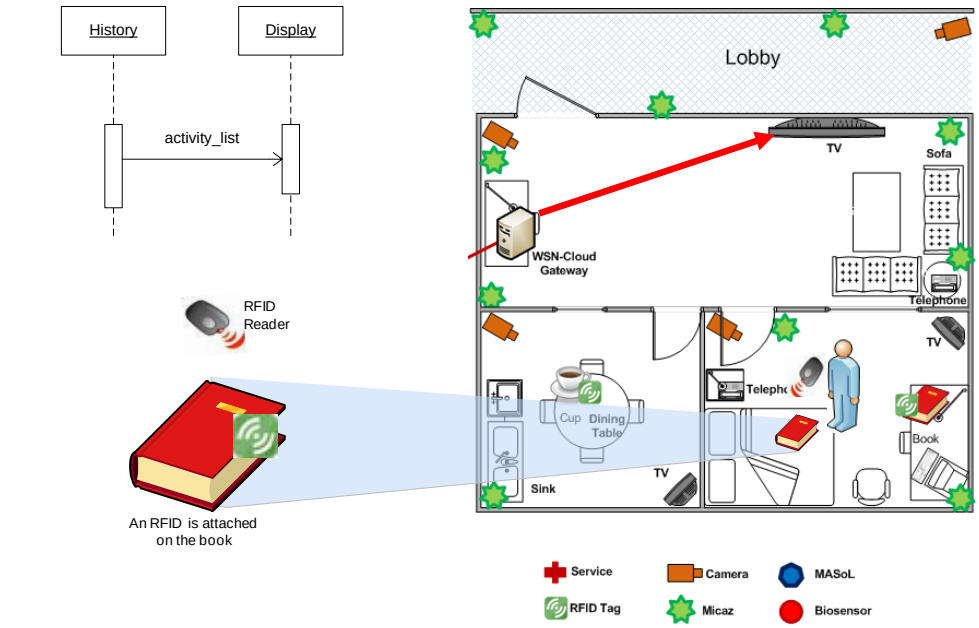


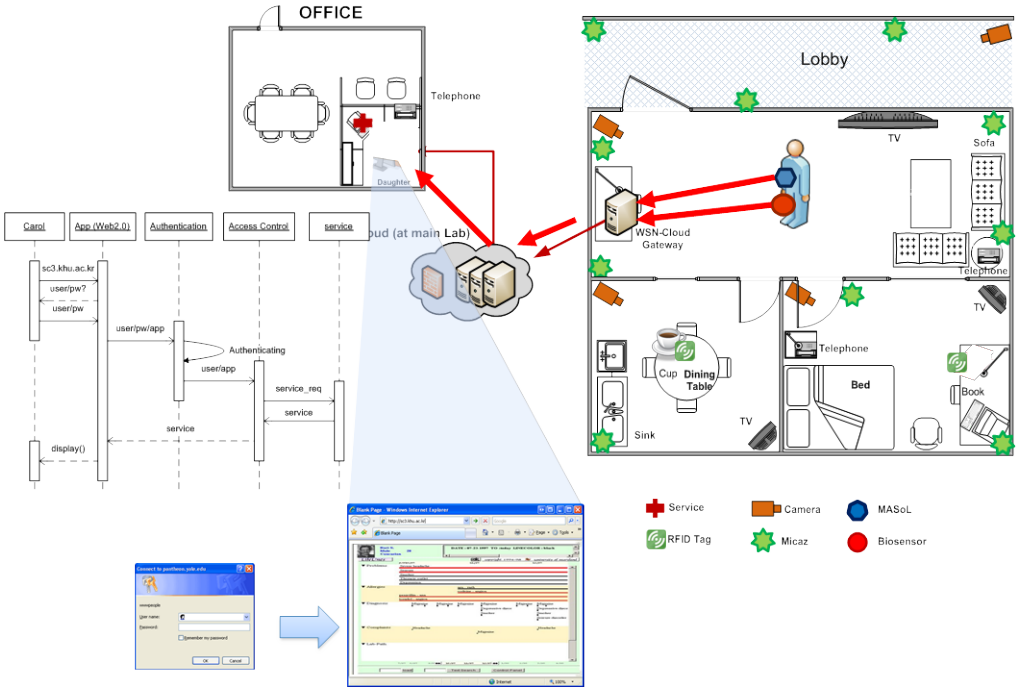
Correct Movement	Wrong Movement
Standing up	Standing up
Standing down	Standing down
Left hand up	Right hand
Right hand up	Right hand down

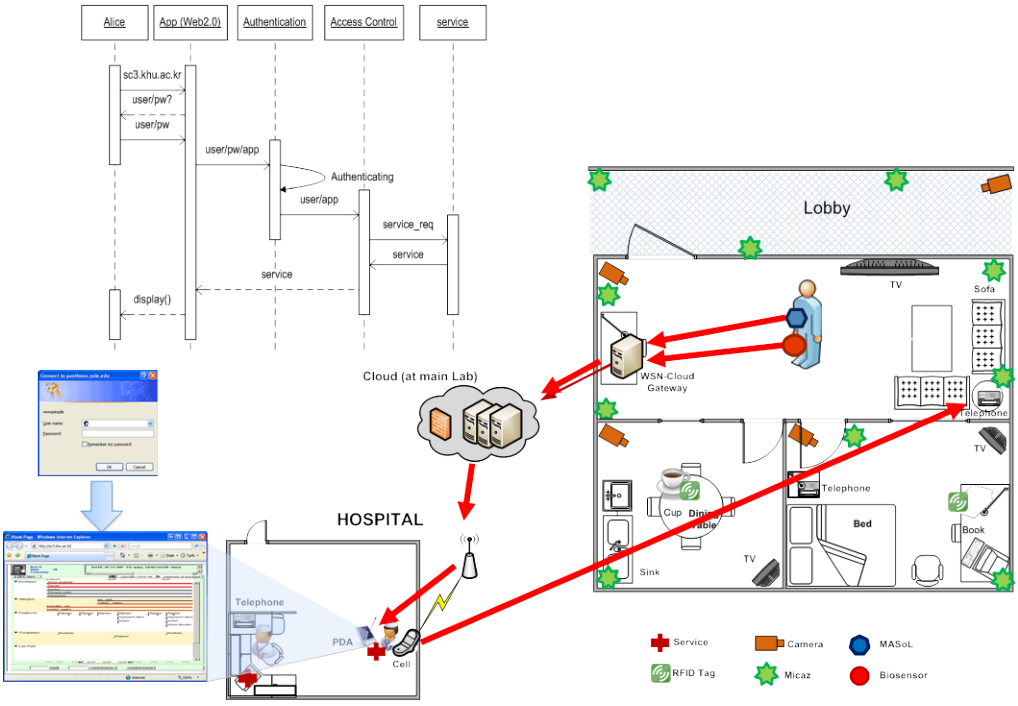
10:00 AM: Looking for the book he was just reading

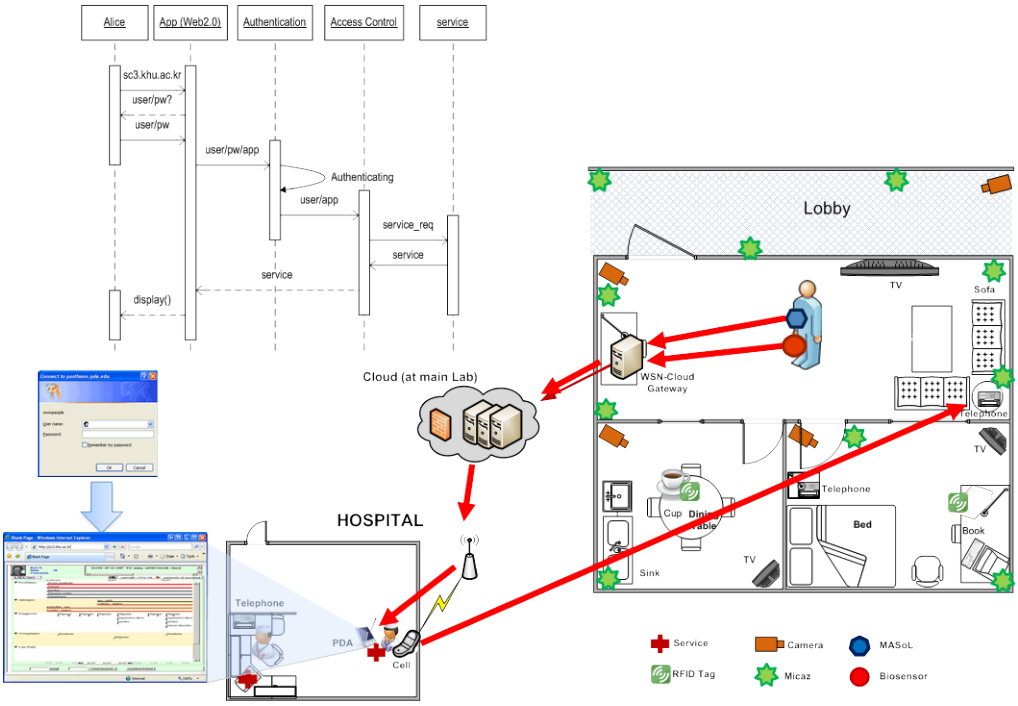


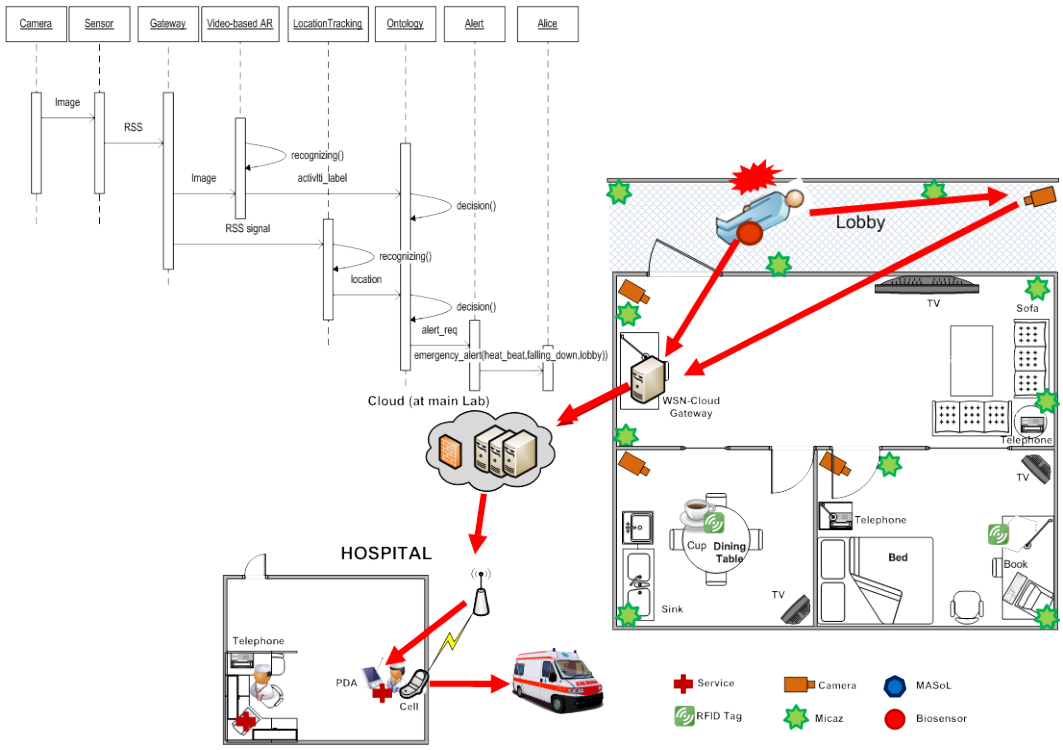
10:00 AM: Looking for the book he was just reading

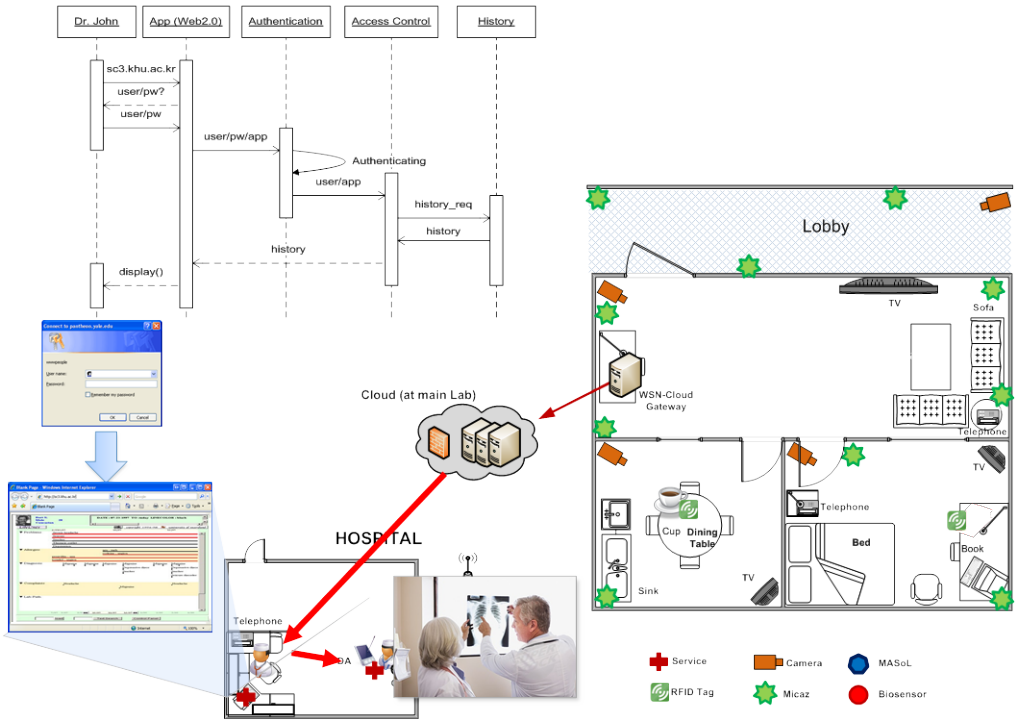



OFFICE
Telephone
Daughter
Lobby
TV
Sofa
WSN-Cloud Gateway
Telephone
TV
Telephone
Cup
Dining Table
Bed
Book
Sink
TV
Carol
App (Web2.0)
Authentication
Access Control
service
oud (at main Lab)
sc3.khu.ac.kr
user/pw?
user/pw
user/pw/app
Authenticating
user/app
service_req
service
service
display()
Service
Camera
MASoL
RFID Tag
Micaz
Biosensor


Alice
App (Web2.0)
Authentication
Access Control
service
sc3.khu.ac.kr
user/pw?
user/pw
user/pw/app
Authenticating
user/app
service_req
service
service
display()
Cloud (at main Lab)
HOSPITAL
Telephone
PDA
Cell
Lobby
TV
Sofa
WSN-Cloud Gateway
Telephone
TV
Telephone
Cup
Dining Table
Bed
Book
Sink
TV
Service
Camera
MASoL
RFID Tag
Micaz
Biosensor


Alice
App (Web2.0)
Authentication
Access Control
service
sc3.khu.ac.kr
user/pw?
user/pw
user/pw/app
Authenticating
user/app
service_req
service
service
display()
Cloud (at main Lab)
HOSPITAL
Telephone
PDA
Cell
Lobby
TV
Sofa
WSN-Cloud Gateway
Telephone
TV
Cup
Dining Table
Telephone
Bed
Book
Sink
TV
Service
Camera
MASoL
RFID Tag
Micaz
Biosensor


Camera
Sensor
Gateway
Video-based AR
LocationTracking
Ontology
Alert
Alice
Image
RSS
recognizing()
Image
activity_label
decision()
RSS signal
recognizing()
location
decision()
alert_req
emergency_alert(heat_beat,falling_down,lobby))
Cloud (at main Lab)
HOSPITAL
Telephone
PDA
Cell
Lobby
TV
Sofa
WSN-Cloud Gateway
Telephone
TV
Cup
Dining Table
Telephone
Bed
Book
Sink
TV
Service
Camera
MASoL
RFID Tag
Micaz
Biosensor



11.6 Scenario Deployment at ITRC Test-bed Room

This section presents the primary results of our SC^3 development. A part of system has implemented including sensor networks, a gateway between home network and Cloud, and a Cloud server. On the Cloud, we have implemented Authentication, Access Control, Activity Recognition, Ontology Engine and a Web 2.0 interface for doctors, nurses, and the patient to access patient information.

A home network is deployed with sensors and cameras to detect user's activity as illustrated in Figure 68 . We customized our ITRC (Information Technology Research Center) test-bed room as a patient's home environment with a living room, a kitchen, and a bedroom. We use a WiTilt V3 sensor supported accelerometer and gyroscope attaching on the patient's right hand to detect his activity such as taking medicine, reading book, eating, teeth brushing. In each room, we deploy a TinyOSMall PIR motion sensor to detect if the patient is in the room. A Logitech wide-angle web camera is attached on the wall of the living room and the kitchen to detect his movement such as watching TV, doing exercise. The home gateway is deployed at the patient's home to collect and transmit raw data from sensors and cameras to the Cloud. We installed a free source code Enomaly ECP in 4 PCs Pentium IV dual-core 2.5GHz, 3GB RAM to serve as a Cloud server.

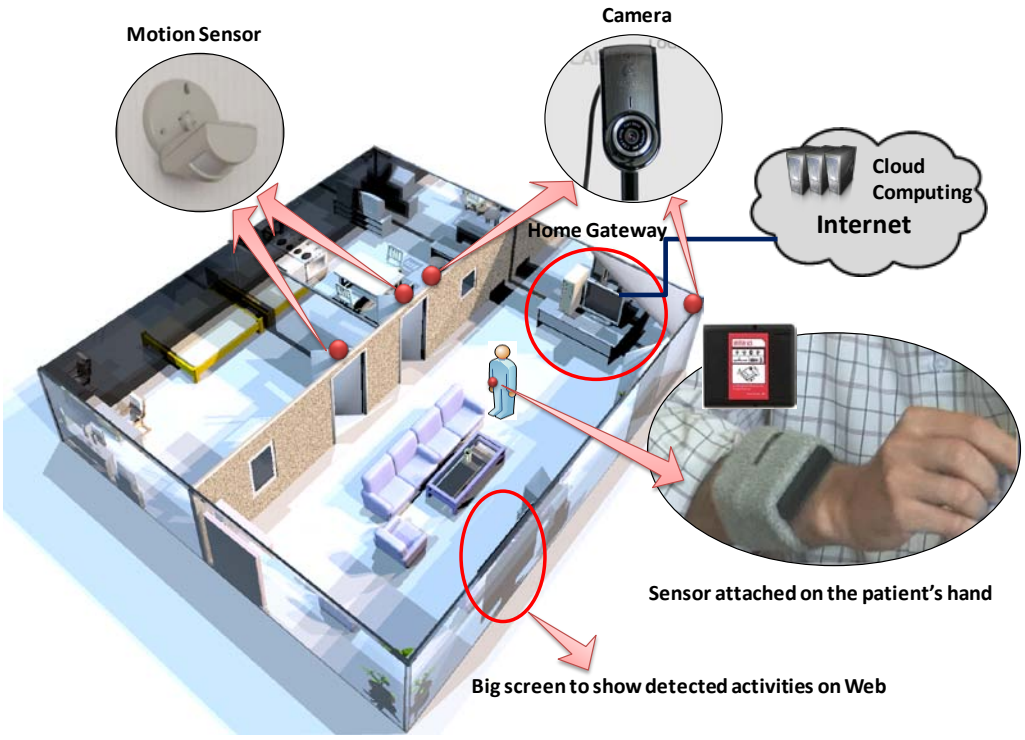


Figure 68 SC^3 deployment at ITRC test-bed room.

A sample scenario is implemented in order to show how SC^3 supports an Alzheimer's disease patient. It works as follows.

At 7 o'clock in the morning, the patient enters the kitchen and has breakfast. When he enters, the motion sensor sends a sensed signal to the SC^3. SC^3 detects he is in the kitchen, so it sends a command to turn on the light. While he is waiting for breakfast, he sits on the chair and looks at the TV. SC^3 detects his posture by collecting image data from the camera and inferring the activity. So it sends a command to turn on the TV. Then, SC^3 collects gyroscope and accelerometer signal from the embodied sensor and infers his eating and teeth brushing action (Figure 69).

Figure 69 SC^3 recognizes the patient is watching TV, so it turns on the TV. It also detects he is eating, and teeth brushing.

After breakfast, the patient reads a book in the bedroom. Detecting that the patient is reading, SC^3 turns off the TV so that he can focus on the book. A while later, SC^3 recognizes that he did not take medicine and do exercise for today by checking the activity database which it has recorded. So it sends a sound reminder "Take medicine please!", and then "Take exercise please!" to him. When the patient performs those two actions, it updates to the database so that it will not reminder him later on (Figure 70).

After doing exercise, the patient wants to read the book that he was reading before. However, he forgets where he left it. So he looks at the big screen on the wall which displays all activities he has been performing. It is a website showing Time, Activity, and Location (Figure 71). He then finds down the last time he was reading the book is in the bedroom, so he can easily gets the book. The patient also checks his health condition and sends a report to Cloud through a Web 2.0 interface.

Figure 70 SC^3 detects he is reading, and reminds him to take medicine and do exercise. It then records his actions to the database and not remind later on.

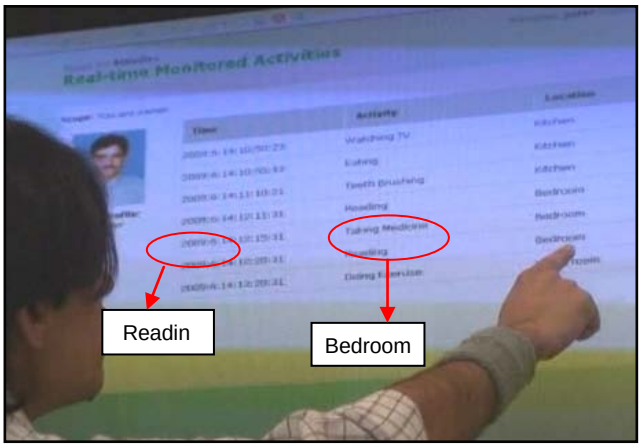


Figure 71 A big screen shows all activities.

At the hospital, the nurse accesses to the Cloud and checks the patient's health condition. She also can see whether he forgot to do something such as taking medicine, doing exercise. As she concludes the patient is not getting better, she comes to the doctor and they have a short discussion. After that, the doctor adds a new medication and lets the nurse brings to the patient (Figure 72).

Figure 72 At the hospital, nurse and doctor check the patient condition via Cloud. A new medication is added and brought to the patient.

Chapter 12

CONCLUSION AND FUTURE WORK

12 Conclusion and Future Work

12.1 Conclusion

This paper introduces Secured WSN-integrated Cloud Computing for u-Life Care, called SC^3. It provides a number of featured components, including security and privacy control, WSN-Cloud integration mechanism, dynamic collaboration between Clouds, and an activity recognition engine to enable many u-Life care services. We also present our primary result of development, and then discuss about its potentialities and benefits.

12.2 Future Work

There are still many works ahead. The first future work that we plan to work on is to provide more services to different kinds of patient's disease such as stroke, Parkinson disease, etc. The number of activities will be increased to support more services. A number of wireless medical sensors are under developed. They will be used to collect health data of patient seamlessly. We also will focus more on security and privacy for Cloud Computing. Currently, most users do not want to store their personal health data on Clouds because it is not safe and reliable. Another work is to extend our development into various such as manufacturing, military services.

References

[1] Korea u-Life care system

[2] Microsoft HealthVault http://healthvault.com

[3] Google Health https://www.google.com/health

[4] Unified Cloud Interface Standardization http://code.google.com/p/unifiedcloud/

[5] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, and David E. Culler. SPINS: security protocols for sensor networks. Wireless Networks, 8(5):521–534, 2002.

[6] Sencun Zhu, Sanjeev Setia, and Sushil Jajodia. LEAP: efficient security mechanisms for large-scale distributed sensor networks. In Proc. of the 10th ACM Conf. on Computer and Comm. security, pages 62–72, NY, USA, 2003.

[7] Chris Karlof, Naveen Sastry, and David Wagner. TinySec: a link layer security architecture for wireless sensor networks. In Proc. of the 2nd Int. Conf. on Embedded networked sensor systems, pages 162–175, Baltimore, MD, USA, November 2004.

[8] Taejoon Park and Kang G. Shin. LiSP: A lightweight security protocol for wireless sensor networks. Trans. on Embedded Computing Sys., 3(3):634–660, 2004.

[9] Erik-Oliver Bla and Martina Zitterbart:. Towards acceptable public-key encryption in sensor networks. In proc. of 2nd International Workshop on Ubiquitous Computing, pages 88–93, Miami, USA, 2005.

[10] S. Ganeriwal and M. B. Srivastava, “Reputation-based framework for high integrity sensor networks,” in Proc. of ACM Security for Ad-hoc and Sensor Networks, Oct. 2004, pp. 66–67.

[11] A. Boukerche, X. Li, and K. EL-Khatib, “Trust-based security for wireless ad hoc and sensor networks,” Computer Comm., vol. 30, pp. 2413–2427, Sept. 2007.

[12] Z. Yao, D. Kim, and Y. Doh, “PLUS: Parameterized and localized trust management scheme for sensor networks security,” in Proc. of the 3rd IEEE Int. Conf. on Mobile Ad-hoc and Sensor Systems, Vancouver, Canada, Oct. 2006, pp. 437–446.

[13] K. Liu, N. Abu-Ghazaleh, and K.-D. Kang, “Location verification and trust management for resilient geographic routing,” J. of Parallel and Distributed Computing, vol. 67, no. 2, pp. 215–228, 2007.

[14] H. Chen, H. Wu, X. Zhou, and C. Gao, “Reputation-based trust in wireless sensor networks,” in Proc. of Int. Conf. on Multimedia and Ubiquitous Engineering, Korea, Apr. 2007, pp. 603–607.

[15] R. A. Shaikh, H. Jameel, S. Lee, S. Rajput, and Y. J. Song, “Trust management problem in distributed wireless sensor networks,” in Proc. of 12th IEEE Int. Conf. on Embedded Real Time Computing Systems and its Applications, Sydney, Australia, Aug. 2006, pp. 411–414.

[16] M. Momani, S. Challa, and K. Aboura, “Modelling trust in wireless sensor networks from the sensor reliability prospective,” in Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications, T. S. et al., Ed. Springer, 2007, pp. 317–321.

[17] S. Buchegger and J.-Y. L. Boudec, “Self-policing mobile ad hoc networks by reputation systems,” IEEE Communications Magazine, vol. 43, no. 7, pp. 101-107, July 2005

[18] Matsumoto, T., Imai, H., “Human Identification through Insecure Channel”, Advances in Cryptology - EUROCRYPT 91, Lecture Notes in Computer Science, Springer-Verlag 547 (1991), pp. 409—421

[19] Wang, C.H., Hwang, T., Tsai, J.J., “On the Matsumoto and Imai's Human Identification Scheme”, Advances in Cryptology - EUROCRYPT 95, Lecture Notes in Computer Science, Springer-Verlag 921 (1995) 382—392

[20] Matsumoto, T., “Human-computer cryptography: An attempt”, 3rd ACM Conference on Computer and Communications Security, (1996) 68—75

[21] Xiang-Yang Li, Shang-Hua Teng, “Practical Human-Machine Identification over Insecure Channels”, Journal of Combinatorial Optimization, 3 (1999) 347—361

[22] Hopper, N.J., Blum, M., “Secure Human Identification Protocols”, Advances in

Cryptology - Asiacrypt 2001, Lecture Notes in Computer Science, Springer-Verlag 2248 (2001) 52—66

[23] Shujun Li, Heung-Yeung Shum, "Secure Human-computer identification against Peeping Attacks (SecHCI): A Survey", Unpublished report, available at Elsevier's Computer Science Preprint Server, (2002).

[24] Daphna Weinshall, "Cognitive Authentication Schemes Safe Against Spyware", IEEE Symposium on Security and Privacy, (2006) 295—300

[25] Philippe Golle and David Wagner, "Cryptanalysis of a Cognitive Authentication Scheme", Cryptology ePrint Archive, Report 2006/258, http://eprint.iacr.org/

[26] Rachna Dhamija, Adrian Perrig, "Deja Vu: A user study using images for authentication", Proc. of the 9th USENIX Security Symposium, (2000) 45—58

[27] Passfaces Corporation, "PassfacesTM; The science behind PassfacesTM", Visit http://www.passfaces.com (2005).

[28] Sorensen, "PassPic - Visual Password Management", Visit http://www.authord.com. (2002).

[29] Bell, D. E., and L. J. LaPadula, "Secure Computer Systems: Mathematical Foundations and Model", Bedford, MA: The Mitre Corporation, 1973.

[30] Lampson, B. W., "Dynamic Protection Structures" AFIPS Conference Proc, 35, 1969, pp. 27–38

[31] Bell, D. E., and L. J. LaPadula, Secure Computer Systems: Mathematical Foundations and Model, Bedford, MA: The Mitre Corporation, 1973

[32] DoD Trusted Computer System Evaluation Criteria (TCSEC), DoD 5200.28-STD.

[33] D. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, R. Chandramouli, Proposed NIST Standard for Role-Based Access Control, ACM Transaction on Information and System Security, Vol. 4, No. 3, 2001, pp. 224-274

[34] Rodríguez MD, Favela J, Martínez EA, Muñoz MA..Location-aware Access to Hospital information and services. IEEE Transactions on Information Technology in Biomedicine 2004.

[35] Motta GH, Furuie SS.. A Contextual Role-Based Access Control Authorization Model for Electronic Patient Record. IEEE Transactions on Information Technology in Biomedicine 2003, pp. 202-207.

[36] Jean Bacon, Ken Moody, Walt Yao. "A model of OASIS role-based access control and its support for active security". ACM Transactions on Information and System Security (TISSEC), Volume 5 Issue 4, 2002.

[37] ITU-T Recommendation X.509, the Directory: Authentication Framework, Int'l Telecomm. Union, Geneva, 2000; ISO/IEC 9594-8.

[38] Qun Ni, Alberto Trombetta, Elisa Bertino, and Jorge Lobo, Privacy-aware Role Based Access Control, Symposium on Access Control Models and Technologies, ACM, New York, NY, USA, 2007, pp. 41 - 50 .

[39] Georgios V. Lioudakis, Eleftherios A. Koutsoloukas, Nikolaos L. Dellas, Nikolaos Tselikas, Sofia Kapellaki, George N. Prezerakos, Dimitra I. Kaklamani, and Iakovos S. Venieris, A middleware architecture for privacy protection, Computer Networks: The International Journal of Computer and Telecommunications Networking, Elsevier North-Holland, Inc., New York, NY, USA, 2007, pp. 4679-4696 .

[40] Paolo Guarda, Nicola Zannone, Towards the development of privacy-aware systems, Information and Software Technology, Butterworth-Heinemann, Newton, MA, USA, 2009, pp. 337-350.

[41] Eugster P. T, Felber P. A, Guerraoui R and Kernmarrec A-M. The many faces of publish/subscribe. In: ACM Computing Surveys, Volume 35, No. 2, pp: 114–131, 2003.

[42] Riabov A, Liu Z, Wolf J, Yu P and Zhang L (2003) New algorithms for content-based publication-subscription systems. In: Proceedings of 23rd International Conference on Distributed Computing Systems ICDCS)

[43] Casalicchio E, Morabito F, Cortese G and Davide F. A Novel Approach to Adaptive Content-based Subscription Management in DHT-based Overlay Networks. In: Journal of Grid Computing, Springer, Volume 4, pp: 343-353, 2006

[44] Casalicchio E and Morabito F. Distributed subscriptions clustering with limited knowledge sharing for content-based publish/subscribe systems. In: Proceedings of Sixth IEEE International Symposium on Network Computing and Applications (NCA), 2007
[45] Carzaniga A, Wolf A. L (2003) Forwarding in a content-based network. In: Proceedings of ACM SIGCOMM. Karlsruhe, Germany, pp: 163–174, August
[46] Wu K, Chen S and Yu P. S (2004) VCR Indexing for Fast Event Matching for Highly-Overlapping Range Predicates. In: ACM Symposium on Applied Computing, Nicosia, Cyprus
[47] Liu Z. Parthasarthy S, Ranganathan A, Yang H (2007) Scalable Event Matching for Overlapping Subscriptions in Pub/Sub Systems. In: Proceedings of Distributed Event Base Systems (DEBS), ACM Press, Toronto, Canada.
[48] S. Nepal, J. Chan, S. chen, D. Moreland, and J. Zic. "An Infrastructure Virtualisation SOA for VNO-based Bussiness Models". IEEE SCC 2007, July 2007, pp-41-51
[49] S. Nepal, and J. Zic. "A Conflict Neighbouring Negotiation Algorithm for Resource Services in Dynamic Collaboration", Journal, Publisher, Location, Date, pp. 1-10.
[50] S. Nepal, J. Zic, and J. Chan. "A Distributed Approach for Negotiating Resource Contributions in Dynamic Collaboration". PDCAT 2007, pp. 82-86. 3-6 Dec 2007.
[51] S. Chen, S. Nepal, C. Wang, and J. Zic. "Facilitating Dynamic Collaborations with eContract Services" 2008 IEEE International Conference on Web Services.
[52] J.K. MacKie-Mason, and H.R. Varian. "Generalized Vickrey Auctions". Working paper, University of Michigan, 1994.
[53] K. Bubendorfer, and W. Thomson. "Resource Management Using Untrusted Auctioneers in a Grid Economy". In Proceedings of the Second IEEE International Conference on e-Science and Grid Computing (e-Science'06).
[54] http://www.informationweek.com/news/services/storage/showArticle.jhtml?articleID=212201778&subSection=Security
[55] http://cloudsecurity.org/
[56] Y. Ivanov, A. Bobick, Recognition of visual activities and interactions by stochastic parsing, IEEE Trans. Patt. Anal. Mach. Intell. 22 (8) (2000) 852–872.
[57] S. Park, J.K. Aggarwal, Semantic-level understanding of human actions and interactions using event hierarchy, in: CVPR Workshop on Articulated and Non-Rigid Motion, Washington, USA, 2004.
[58] N. Robertson, I. Reid, Behaviour understanding in video: a combined method, in: ICCV, China, 2005.
[59] T. Xiang, S. Gong, Beyond tracking: modeling action and understanding behavior, Int. J. Comp. Vis. 67 (1) (2006) 21–51.
[60] M. Yamamoto, H. Mitomi, F. Fujiwara, T. Sato, Bayesian classification of task-oriented actions based on stochastic context-free grammar, in: Int. Conf. Auto. Face Ges. Recog., UK, 2006.
[61] M. Kass et al., Snakes: active contour models, Int. J. Comp. Vis. 1 (4) (1988) 321-331.
[62] T. Chan and L. Vese, Active contours without edges, IEEE Trans. Image Proc. 10 (2001) 266-277.
[63] Z-J. Wang, X-F. XU et al. Genetic Algorithms for collaboration cost optimization-oriented partner selection in virtual enterprises. International Journal of Production Research, Vol.47, No. 4, 15 February 2009.
[64] C. L. Hwang, and K. Yoon. Multiple attribute decision making: Methods and applications. Berlin: Springer, 1981.
[65] Z. Fuqing; H. Yi and Y. Dongmei. A multi-objective optimization model of the partner selection problem in a virtual enterprise and its solution with genetic algorithms. Int J. of Adv. Manufacturing Technology, Vol. 37:1220, 2008.
[66] F. Cheng, F. Ye and J. Yang. Multi-objective optimization of collaborative manufacturing chain with time-sequence constraints. Int J Adv Manuf. Technol, 2009, 40:1024–1032.
[67] R. Cowan, N. Jonard et al. Bilateral collaboration and the emergence of innovation networks. Management Science, 53, 1051–1067, 2007.

[68] S. Castano, A. Ferrara, G. Hess, “Discovery-Driven Ontology Evolution”. The Semantic Web Applications and Perspectives (SWAP), 3rd Italian Semantic Web Workshop, PISA, Italy, 18-20 December, 2006.

[69] P. Cimiano, P. Haase, Q. Ji, T. Mailis, G. Stamou, G. Stoilos, D. T. Tran, V. Tzouvaras, “Reasoning with Large A-Boxes in Fuzzy Description Logics using DL reasoners: An Experimental Evaluation” In Proceedings of the ESWC Workshop on Advancing Reasoning on the Web: Scalability and Commonsense. 2008.

[70] G. Flouris, and D. Plexousakis, “Handling Ontology Change:Survey and Proposal for a Future Research Direction”, Institute of Computer Science, FORTH, Heraklion, Crete, Greece, September 2005.

[71] G. Flouris, D. Plexousakis, and G. Antoniou, “A Classification of Ontology Changes”, In the Poster Session of Semantic Web Applications and Perspectives (SWAP), 3rd Italian Semantic Web Workshop, PISA, Italy, 2006.

[72] E. Gahleitner, W. Behrendt, J. Palkoska, and E. Weippl, “On cooperatively creating dynamic ontologies,” in Proceedings of the 16th ACM Conference on Hypertext and Hypermedia. Salzburg, Austria: ACM Press, September 2005, pp. 208–210.

[73] Gruber, T. "A Translation Approach to Portable Ontology Specifications", Knowledge Acquisition , pp 199-220, 1993.

[74] M. Klein. “Change Management for Distributed Ontologies”, PhD Thesis, Department of Computer Science, Vrije University, Amsterdam, 2004.

[75] P. Shvaiko and J. Euzenat, “ Ten Challenges for Ontology Matching”, In Proceedings of The 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE), August 2008.

[76] M. Singh, and M. Huhns, “Service Oriented Computing: Semantics, Processes, Agents”, John Wiley & Sons, West Sussex, UK, 2005.

[77] B. Smith, Blackwell Guide to the Philosophy of Computing and Information, ser. Blackwell Philosophy Guides. Blackwell Publishing, October 2003.

[78] T. Strang, C. Linnhoff-Popien, “A Context Modeling Survey”, In: Workshop on Advanced Context Modelling, Reasoning and Management, The Sixth International Conference on Ubiquitous Computing (UbiComp), Nottingham/England, 2004