

A Context-aware Healthcare Architecture For The Elderly

OYEKANMI TOLULOPE PETER

A Thesis
In
The Department
Of
Computer Science And Software Engineering

Presented In Partial Fulfillment Of The Requirements
For The Degree Of Master Of Software Engineering
Concordia University
Montreal, Quebec, Canada

December 2015

© Oyekanmi Tolulope Peter, 2015

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Oyekanmi Tolulope Peter

Entitled: A Context-Aware Healthcare Architecture for the Elderly

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Software Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final Examining Committee:

_____Chair
Dr. E. Shihab

_____Examiner
Dr. N. Shiri

_____Examiner
Dr. Y. Yan

_____Supervisor
Dr. V. S. Alagar

Approved by _____
Chair of Department or Graduate Program Director

Dr. Amir Asif
Faculty of Engineering and Computer Science

Date _____

Abstract

In order to provide dependable healthcare services for the elderly, it is necessary to have a patient-centric healthcare architecture in which context-aware healthcare services can be provided at any time and anywhere. Such a service automation has the virtues to overcome the disadvantages arising from the disabilities that are inherent in the elderly population, physically challenged, and those who live in remote areas. In order that patients trust the healthcare services provided by the system, the creation of healthcare services must be founded on accurate personalized health model of patients, and must be delivered by experts through dependable medical devices and secure channels. Motivated by this goal, this thesis proposes a layered health model that can be personalized to meet the privacy requirements of a patient, and constructs a context-aware healthcare architecture in which healthcare services for each patient is specialized based on personalized health models, health contexts, and emerging health situations. A prototype implementation of the architecture is validated for Hypertension and Dementia case studies.

Acknowledgments

I would love to appreciate all those involved in ensuring the successful of this thesis. First, I would like to express my sincere gratitude to my supervisor Prof. Vangalur Alagar for his patience and motivation, his wide knowledge and perspective had profoundly helped me in completing this research. I would also like to extend my gratitude to Dr. Mubarak Mohammad for his help which was truly indispensable for this work. Finally, I would like to dedicate this work to every other person that has helped in completing this research.

Contents

List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 Research Contributions	4
2 Related Work	5
2.1 Vulnerabilities Of Elderly	5
2.2 Healthcare Needs Of Elderly	6
2.3 Healthcare Legal Acts	8
2.3.1 HIPAA	8
2.3.2 Ontario Health Information Protection Acts	8
2.3.3 Saskatchewan Health Information Protection Act	8
2.3.4 Alberta Health Information Protection Act	9
2.3.5 Nationwide Privacy and Security Framework for Electronic Exchange of Individually Identifiable Health Information	9
2.3.6 Common Framework For Healthcare Information Exchange	9
2.4 Healthcare Systems Research Areas	10
2.4.1 Education and Awareness	10
2.4.2 Helpline	10
2.4.3 Diagnostic and Treatment Support	11
2.4.4 Communication and Training	11
2.4.5 Diseases and Epidemic Outbreak Tracking	11
2.4.6 Remote monitoring	12
2.4.7 Remote data collection	12
2.5 Existing Healthcare Systems	12

3	Context Awareness In Healthcare	16
3.1	Context	17
3.2	Terminologies	18
3.2.1	Health Determinants	18
3.2.2	Health Conditions	20
3.2.3	Health Situations	21
3.2.4	Health Context (HT)	22
3.2.5	Health Adaptation (HA)	22
3.3	Entity Model	23
3.3.1	Elderly Home (EH)	23
3.3.2	Inter-BAN Communication	25
3.3.3	Cloud Cyberspace (CC)	26
3.3.4	Service Providers (SP)	28
3.4	Context-dependent Security and Privacy	29
3.4.1	Common Issues	29
3.4.2	The IAS-octave Model	31
4	Architectural Design	33
4.1	Review Of Existing Context-Aware Architectures	33
4.2	Proposed Architecture	35
4.2.1	Architecture Overview	36
4.2.2	Data Store Module	36
4.2.3	Sensor Actuator Network (SAN)	36
4.2.4	Controller Module (CM)	37
4.2.5	Context Manager Module	38
4.2.6	Situation Manager Module	38
4.2.7	Adaptation Manager Module	39
4.2.8	Significance Of Architecture	39
4.3	Process Flow	40
5	Detailed Design	42
5.1	Data Interfaces	42
5.1.1	ISensor	42
5.1.2	ISensorReading	43
5.1.3	IContext	43
5.1.4	IProvider	44
5.1.5	IBiodata	45

5.1.6	IGuardian	46
5.1.7	IPolicy	46
5.1.8	ISituation	46
5.1.9	IAdaptation	47
5.1.10	IHealthModel	47
5.2	Module Interfaces	48
5.2.1	IContextManager	48
5.2.2	ISituationManager	49
5.2.3	IAdaptationManager	49
5.2.4	IController	49
5.3	Helper Interfaces	50
5.3.1	ILogger	50
6	Implementation	51
6.1	Implementation Platform	51
6.1.1	JDK Characteristics	52
6.2	CAHA Implementation	52
6.3	Case Studies	53
6.3.1	Dementia Patient Case Study	53
6.3.2	Hypertension Patient Case Study	55
6.4	Tests And Results	57
7	Conclusion And Future Work	59
7.1	Limitations	60
7.2	Future Work	61
8	Bibliography	62
	Appendix A Situation Expression Language	71
	Appendix B Workflow and Policy Expression Language	76
	Appendix C Pseudo code	80

List of Figures

2.1	Ubimon System Architecture [62]	13
2.2	MobiCare System Architecture [62]	14
3.1	Health Determinants	18
3.2	Life Expectancy In Years	20
3.3	Entity Model	25
4.1	The Three-Tiered Formalism of Context-Aware System [108]	34
4.2	CAHA	35
4.3	Process Flowchart	40
5.1	Sequence Diagram	48

List of Tables

- 1.1 Proportion of population aged 65 years and over, selected countries [73] 2
- 3.1 Comparison of cloud service models [4] 27

Chapter 1

Introduction

Providing healthcare for all, especially for elderly and physically challenged people, at any place and at anytime is being recognized as the most efficient and cost-effective way in spite of the many challenges that are to be overcome in making it dependable. The challenge lies in seamlessly integrating communication technology, and information and networking technology with social aspects. It is in this context that this thesis makes a contribution, through the introduction of a context-aware healthcare architecture for providing healthcare services for the elderly population.

The word “Elderly” is descriptive of someone who has reached old age. Despite that there is no consensus on an accepted and acceptable definition of old age, most developed countries accept the range of 60 to 65 as the inception of old age[114]. This also agrees with a study by Norman Ryder[95] in 1975 which pegs the inception of old age at 65 on the basis that a person aged 65 has a remaining life-expectancy of 10 years. In Canada, this is the age when many people are eligible for social services such as government pensions[32]. Even at a time, some provinces in Canada made it compulsory for their workers to retire at this age[32]. However, a survey average in [114] revealed that most US citizens believe that old age actually starts at 68 with some of them going as high as 74 as being the ideal age. In general, women believed that anybody aged 70 and above can be termed as an elderly while men agreed on age 66 as the starting age according to [114]. Generally, the “Elderly” population can be classified into “young-old” (65 to 74 years of age), the “aged” (75 to 84 years of age), and the “oldest-old” (more than 85 years of age).

Between 2000 and 2010, the population of the elderly throughout the world increased at 15.1% which is faster than the rate the total U.S population was growing [27] with a U.S Census in 2010 reporting that there were 40.3 million elderly people in the United States which is far different from the 3.1 million reported in 1900. In 2012, the population of the elderly made up 11% of the world’s population and was projected to reach 22% by 2050, with 68% of the world’s population over 80 living in Asia and Latin America and the

Country And Rank	2005%	2050%
1-Japan	20.0	39.6
2-Italy	19.6	33.7
3-Germany	18.9	29.6
22-Canada	13.1	26.3
23-Australia	13.1	25.7
24-United States	12.4	20.6
25-New Zealand	12.1	26.2
28-Korea	9.1	38.2
29-Turkey	5.9	17.0
30-Mexico	5.3	21.1

Table 1.1: Proportion of population aged 65 years and over, selected countries [73]

Caribbean[52]. As at 1998, old people in Canada were estimated at 3.7 million which is more than the 2.4 million estimated in 1981 and about twice of the number in 1921 [53]. Between 1986 and 2010, the number and proportion of elderly people in Canada increased from 2.7 million to 4.8 million and from 10% to 14% of the population [32]. [73] reports that Canada’s population growth rate is higher than Japan and Korea because a higher value of total fertility rate and higher immigration rate as it has been evidenced that 28.4% of the population now aged 65+ in Canada were born outside Canada. At the moment, one in six Europeans can be regarded as elderly [101]. In Australia, older people currently make up 13.6% of the population[91]. According to [73], Japan is the demographically oldest country in the world with 20% of its population aged 65+. It is estimated that by 2020, China will have 230 million elderly people, thereby making it the largest population of elderly worldwide, while India will come next with about 158 million old people [44]. This study also asserts that by the year 2030, the number of people aged 18 across the world will be lower than the ones over age 65, with the number of those aged 85 and above being 8.5 million [37]. [43] further projects that the population of the elderly in the United States will more than double to about 71 million by 2030. As shown in Table 1.1, Korea, in 2005 had 9.1% of its population aged 65+ but is expected to almost catch up with Japan in 2050. Further, by 2050, out of every four persons, one will be an elderly [31] with life expectancy reaching 75[52].

Due to the projected growth in the population of the elderly, it is expected that the proportion of care givers assisting the elderly will not be sufficient and the cost of giving care for the elderly will also be on the increase. A poll in 2010 revealed that about 70% of Canadians hold the notion that the health system must be changed in order to meet their needs as they age, while 72% of Canadians believe they won’t be able to maintain their health as they grow older [31]. In [25], it is further reported that the world is in need of around 2.4 million medical experts in order to be able to successfully meet healthcare needs. The reports [31, 43] predict that Canadian healthcare costs is expected to increase by 26% between 2010 and 2036 as a result of population aging, the projected increase being 25% by 2030. An effective solution to this problem is “healthcare service automation”, which can provide services whenever and wherever they are

demanded thereby maximizing health service utilization and minimizing service cost. This thesis is focused on creating an architecture for implementing such a healthcare system from bottom up.

By healthcare, we not only refer to the treatment phase but also the preventive and curative phase where medical procedures can be initiated to enhance a person's well-being[26]. In recent times, the healthcare industry has used technology to provide better healthcare services to patients. For example, doctors can now monitor patients remotely, track prescriptions as well as communicate with patients without having to be physically present with the patient. A study revealed that computerized physician order entry with decision support reduced the serious medication error rate by 55% [39]. The potential of what can be achieved as healthcare continues to adopt technology has served as an encouragement to venture capitalists who invested \$955 million into health care IT in 2013 [86]. It is also of no surprise that the Canadian Government invested close to \$900 million dollars between 2007 and 2009 in an effort to implement a national health information system [69].

The elderly population must be an important focus area due to the numerous vulnerabilities and healthcare needs they face, especially in providing personal care in their daily living which could range from caring for their personal hygiene to feeding. An elderly person's nutritional requirements must be met, otherwise it could result in malnutrition and degrading health conditions [11]. A health condition that can result out of poor nutrition is diabetes which has been found to lead to higher rates of sudden death. Stroke is another common problem that can arise and co-exist with diabetes [13]. Since sedentary lifestyle, social isolation, loneliness, or depression can lead to malnourishment, and depression medications can also change how nutrients are absorbed or how food tastes, caring the dietary needs of elderly is paramount. For the chronically ill elderly, for example those suffering from arthritis [84] who also have a recurring episode of osteoarthritis [84], care needs to be given in multiple dimensions. It has been reported that about 20% of those who are 55 years and older experience mental disorders such as anxiety disorders, severe cognitive impairment and mood disorders which are not part of normal aging[107]. Mental illness is very difficult to address as many seniors are either unwilling or unable to report their situations. Most elderly people lack knowledge about the causes or symptoms behind their health problems and assume their health problems are simply due to their aging [36]. The survey [44] has revealed that 96% of the elderly have never utilized any of the geriatric welfare services due to lack of awareness. The elderly also go through physical, psychological and financial vulnerabilities [33], become dependent on others which exposes them to societal dangers [64] such physical and psychological assaults, and financial exploitation [105]. The different vulnerable situations and the health conditions discussed above can be related to specific contexts thereby creating a need for a context-aware solution.

1.1 Research Contributions

This research is an applied research that incorporates theoretical analysis and practical application of science. It contains a literature review of an extensive review of reports, articles and books on smart living for the elderly with concentration on security and privacy. A survey of existing architecture solutions is included. The literature review is a key part of this research because without healthcare domain knowledge a healthcare system cannot be automated. The extensive list of references have helped in extracting requirements and specifications for the modeling and development of the conceptual model. We were also able to propose a detailed architecture for the conceptual model.

The following is a list of contributions of this thesis.

- A detailed review of the numerous daily needs and vulnerabilities of the elderly in order to motivate the need for patient-centric and privacy-enhanced health model for the elderly.
- A comprehensive survey of existing healthcare solutions and models for the elderly, legal standards relating to the collection, processing and storage of electronic healthcare records.
- A layered health model, the health determinants which give rise to health contexts of patients. Context-aware healthcare is defined through the definitions of health contexts, situations, and adaptations.
- Context-aware privacy and security solutions.
- An entity model in which elderly (homes) and healthcare providers are linked through a cloud cyberspace.
- A context-aware healthcare architecture based on the entity model, its component-based design and its prototype implementation.

The rest of this thesis is organized as follows: Chapter 2 contains literature survey of related work. Chapter 3 provides a comprehensive review of how healthcare systems using context awareness. Chapter 4 provides details about the context-aware architecture. Chapter 5 contains the full documentation of the detailed design. Chapter 6 describes the case study implementation and benchmarks for the architecture proposed in Chapter 5. Finally, Chapter 7 presents the conclusions of this research and the future work.

Chapter 2

Related Work

In this chapter, we present a comprehensive account of published research and reports in the area of elderly healthcare, in particular on systems that use context-awareness. We present vulnerabilities and healthcare needs of elderly patients in Sections 2.1 and 2.2. A review of healthcare legal acts is presented in Section 2.3. Section 2.4 is about key focus areas of electronic healthcare systems. We conclude the chapter with a review of existing context-aware systems for the elderly in Section 2.5.

2.1 Vulnerabilities Of Elderly

The elderly in society suffer different vulnerabilities today although most of them don't report these due to guilt, embarrassment and shame. As revealed by a study, only 1% to 12% of elderly persons report some form of maltreatment or abuse [105]. Abusers usually start by identifying an area of vulnerability in a victim such as depression or loneliness after which they attempt to build trust with victims by pretending to offer services or care. Once the victim has fallen prey, they proceed to exploit the patient. The three most cited types of abuse old people pass through are physical, psychological and financial [33].

Elderly people are more likely to be dependent on others for special needs such as protection from abuse and exploitation resulting in frailty which further exposes them to societal dangers [64]. Frailty can be defined as experiencing a progressive physiologic decline in multiple body systems, loss of body functions, loss of physiologic reserve, and increased vulnerability to disease and death [41]. Frailty can also be evident by inactivity combined with low energy intake or weight loss or low body mass index [30]. Caring for elderly people suffering frailty often requires extreme medical intervention [41].

The elderly in the society are often targets for fraudulent business practices due to their lack of market knowledge as most of the elderly don't have clear guidance on the appropriate channels to contact for

financial advice [58]. In addition to this, the frail elderly can also be more vulnerable to property damage owing to lack of insurance[93]. According to NARCEA in 1988, on the basis of a study of data from 24 states, 20% of the Americans who were victims of elderly abuse were victims of financial exploitation[105]. A Canadian study published in 1992 reported that more than 50% of all reported cases of elder abuse involved material exploitation [40].

Older people are also vulnerable to physical assaults ranging from caregivers to strangers due to the frailty they suffer [94]. The limitations old people experience during and after disasters can also not be ignored [64]. For example, some elderly people have difficulties surviving disasters because of diminished sensory awareness [64] and lower rescue aids than their younger counterparts [104]. It is also likely that old people have limited mobility in dangerous situations thereby exposing them to physical harm [16].

There are also social vulnerabilities existing for old people with women rating higher than men [74]. Old people are usually isolated from the rest of the society especially in areas with linguistic barriers which lead to loneliness [16]. A survey revealed that on the average, 32.2% of elderly people feel lonely while only 69.5% of elderly have recreational activity outside the home [44]. Reports say the percentages of elderly people staying alone is 37% for those between the ages of 75 and above and 49% for those aged 85 and above thereby making loneliness a big issue for these people [32]. Loneliness in an elderly person can lead to additional health complication such as depression and anxiety. It is of no surprise that on the average, 55.1% of elderly have sad attitudes towards their lives [44].

2.2 Healthcare Needs Of Elderly

The healthcare needs of the elderly are numerous and often urgent. In 1969, it was reported that elderly persons averaged at seven visits each to clinics, compared with only four visits for children and about five visits for adults[72]. According to the Administration on Aging, around 1.86 million elderly people reported problems with two or more activities of daily living such as taking care of personal hygiene, feeding and in-door mobility. Between 1994 and 1995, patients from age 75 and above are reported to have at least three medical disorders [36]. According to another report [43] about 80% of older adults have at least one chronic condition while 50% have at least two chronic conditions. A survey revealed that about 50% of elderly people experience arthritis, 12% have diabetes and about 10% have had a stroke [97]. The US Centers For Disease Control and Prevention estimates that around 80% of older Americans have at least one chronic condition and 50% have at least two [100].

The most important healthcare need of the elderly is proper diagnosis of their health conditions as most old people simply assume their health problems are simply due to aging [36]. A survey revealed that 96% of the elderly have never utilized any of the geriatric welfare services, 46.3% are not even aware of the existence of these services and 59% had to travel more than 3 kilometers for government supplied health services

for proper medical examination [44]. Even in areas where they seem to have access to sufficient qualified healthcare professionals, lack of insurance could also serve as a barrier [19].

There is also a growing need for aiding the mental health status of the elderly. It has been reported that about 20% of those who are 55 years and older experience mental disorders such as anxiety disorders, severe cognitive impairment and mood disorders which are not part of normal aging [107]. [52] reports that 35.6 million people live with dementia worldwide with the numbers expected to double every 20 years and the number reaching 115.4 million in 2050. It is reported in [43] that nearly 7 million older adults suffer from dementia in the United States alone. It is even expected that by 2030, about 63 million elderly persons living in LMIC will be affected by depression thereby resulting in economic and productivity losses of around US\$16.1 trillion [63]. Women are generally known to have a double rate of depression when compared to men with a higher prevalence rate among seniors in healthcare facilities according to a study done in Edmonton [110]. The prevalence for depression among older adults is estimated to be at least 6% in primary care and 30% in long-term care in HIC [63]. Mental illness is very difficult to address as many seniors are unwilling to report their situations.

It is not unusual for the elderly to have physical healthcare needs, for example arthritis which is the most commonest cause of disability for the elderly [84]. A survey over a one-year period revealed that 10% of people over 55 years have a recurring episode of osteoarthritis with quarter of that being severely disabled [84]. In addition, the elderly have special food and nutritional requirements that if not met can lead to malnutrition and degrading health conditions [11]. A health condition that can result out of poor nutrition is diabetes which has been found to lead to higher rates of premature death and coexisting illnesses such as stroke [13].

Meeting the healthcare needs of the elderly through the traditional method of physical visits to the hospital is becoming more and more difficult especially in developing countries due to its growing inefficiency. Most of the elderly have to wait on long queues when they go to hospitals for medical intervention. In addition, cost of transportation in terms of time and money during each visit is also of concern for favoring the use of mobile healthcare systems. It is therefore of little surprise that the 111th United States Congress enacted the 2009 American Recovery and Reinvestment Act (ARRA) which includes a financial incentive of \$19 billion for willing healthcare providers [98] in order to encourage them to make use of mobile systems in providing healthcare services to their patients. Government institutions in low and middle-income countries have also started indicating their intent in using mobile healthcare systems to achieve MDG through the development and deployment of nation-wide health services made achievable as a result of increased availability, better performance and cheaper mobile data rates especially in the areas of remote health-care monitoring and emergency medical care for the elderly [15]. Health authorities usually use the information collected from these systems for taking definite actions in improving the health conditions of the elderly in lacking regions.

2.3 Healthcare Legal Acts

2.3.1 HIPAA

HIPAA was enacted by the United States Congress in 1996 to protect health insurance coverage for medical workers and establish specific guidelines to follow when processing electronic medical information especially on mobile devices due to the sensitivity of the data being measured and processed in the healthcare industry and the risks involved in any probable breaches [46]. It was further stated that HIPAA dictates “protected health information (PHI)” such as name, phone number, email address, photograph, payment information and any other information that can be linked back to the patient should never be disclosed to or tampered by third party without the express permission of the patient. The act also provides certain guidelines on availability of information to authorized personnel and access control to medical equipments used in managing medical records.

2.3.2 Ontario Health Information Protection Acts

In 2004, the province of Ontario [38] enacted “The Ontario Health Information Protection Act”, which is the aggregation of the “Personal Health Information Protection Act, also known as PHIPA”, and “The Quality of Care Information Protection Act”. It was further explained that the Act suggested that physicians should obtain either express or implied consent before disclosing personal health information to external parties as well as the concept of *lockouts* which is a situation where the patient explicitly restricted the physician from disclosing personal health information even to others directly involved in providing healthcare services to the patient[34]. In addition, the Act provides recommendations for information disclosure relating to family member and the police.

2.3.3 Saskatchewan Health Information Protection Act

Saskatchewan Health Information Protection Act is the standard developed and adopted by the Government of Saskatchewan on September 1, 2003 guarding the rights of individuals and obligations of *trustees* regarding the collection, storage, use, access and disclosure of personal health information in the province of Saskatchewan. According to the Act, Personal health information can be defined as information with respect to the physical or mental health of the individual or about any health service provided to the individual while a trustee includes government institutions, community clinics, district health boards, regional health authorities, ambulance operators and health professionals [102].

Saskatchewan Health Information Protection Act upholds the right of the individual to have full control over the use or disclosure of personal health information about himself or herself and trustees must obtain such consent voluntarily and purposefully from the individual before using or disclosing such information. The

Act further dictates that a trustee in charge of personal health information must establish written actionable policies and procedures to ensure security and privacy of such information. In conclusion, it establishes certain conditions trustees must follow before disclosing health information to information management service providers.

2.3.4 Alberta Health Information Protection Act

Alberta Health Information Protection Act is the standard developed by the Government of Alberta guiding health professionals on handling electronic health information in the province of Alberta. The Act states that custodians of electronic health information must submit a privacy impact assessment to, meet any security requirements established by the department of the Provincial Commissioner and obtain approval for access to the provincial Electronic Health Record (EHR) from the Department [87]. According to the Act, a custodian must ensure its electronic health record information system creates and maintains logs containing the essential information about system access. For example, date, time, and the user who performed the access would be recorded. The Act was enforced as Section 20 of the Health Information Amendment Act, 2009.

2.3.5 Nationwide Privacy and Security Framework for Electronic Exchange of Individually Identifiable Health Information

This security framework is the latest released for use by organizations processing medical health information in the United States that focuses on improving the availability of health information and health care quality. It was prepared by the Office of the National Coordinator for Health Information Technology in the U.S. Department of Health and Human Services on December 15, 2008.

The framework states that individuals should be able to assess their information with a reasonable amount of effort in a readable, electronic inclusive format and should be able to make appropriate changes to such information in cases of errors [106]. It also states that the system should restrict information collection and use to the specified purposes and be as open as possible about such uses, for example individuals should have appropriate knowledge of how the system handles information disclosure of their identifiable health information to designated individuals. It further emphasizes that people involved in the development of such a system are expected to put appropriate safeguards in place to minimize the risks of unauthorized or inappropriate access, use or disclosure.

2.3.6 Common Framework For Healthcare Information Exchange

The Common Framework For Healthcare Information Exchange was created by The Markle Foundation to provide important basis for sharing health information between patients and healthcare providers. It contains

16 policies which shows how various healthcare systems can ensure security and privacy of information [111].

The framework states that designers of healthcare systems should be open about the policies, purposes and disclosure of information as much as possible with the involved individuals having unrestricted access to the control and use of such information [103]. It also states that developer of healthcare systems are responsible for ensuring reasonable security safeguards against risks, as well as legal and financial remedies to address any security breaches or privacy violations.

2.4 Healthcare Systems Research Areas

While multiple authors have attempted to classify healthcare systems into different research areas, this section discusses the seven key themes as proposed by United Nations [14].

2.4.1 Education and Awareness

These kind of healthcare systems are used to raise public awareness such as educating people on important health topics, for example HIV/AIDS. This category also extends to systems that make use of text messaging for health promotion or to alert target groups of health campaigns for example, to enhance sensitization about immunization campaigns. These programmes can be set up either as one-way alerts or interactive tools and can often be downloaded to patients' mobile phones or sent as a series of text messages. Using SMS messages for health campaigns is preferred due to low cost, confidentiality and broad reach [15]. However, there have been certain barriers to using mobile technology for education and awareness such as SMS length restrictions, language barriers, illiteracy, shortage of skilled technical personnel and lack of technical support in rural areas [15].

2.4.2 Helpline

These are the healthcare systems that provide a range of health care services to patient over the telephone. This also extends to healthcare systems that provide quick emergency access to health professionals, for example 911 services in Canada. Usually, trained health professionals are available as call center agents to answer queries from patients. This is currently being capitalized on by low income countries to overcome health care challenges such as shortage of health professionals, cost of service and transportation as well as the lack of sources of reliable information. However, a survey has revealed several limitations in the way the model has been implemented. The operational cost of these call centers are hidden costs that are passed on to the consumers. Even for setting up toll-free number there is a "set up" cost. In addition, personnel and data management costs add up to the operational cost. Unfortunately, customers in low income countries are unable to afford the charges involved [15], and as a consequence this service is not well utilized. To be more specific, a recent study on healthcare systems in Egypt found that patients preferred contacting private

transportation providers despite the availability of a national toll-free telephone service as a result of the involved costs and slow response times [15]. In addition, such systems have inconsistent availability in areas with low mobile coverage.

2.4.3 Diagnostic and Treatment Support

Healthcare systems under this category attempt to achieve treatment compliance, disease eradication, and overcome challenges through the reminder systems usually by voice or SMS. Usually, these kind of systems can be used to support patients with conditions such as diabetes and tuberculosis. Patients are also able to use some of this category of healthcare systems to schedule or attend appointments through voice or SMS which is becoming more rampant in areas where fixed-line telephony is minimal. However, while there is a preference for using SMS for treatment compliance programmes across the globe due to its low cost, the use of these systems does not automatically translate to the improvement of health service metrics such as attendance rates by patients [15].

2.4.4 Communication and Training

Healthcare projects within this category provide access to information in form of health science publications, medical institutions and databases using mobile devices for use by health care workers such as nurses. This is useful in cases where medical support is scarce and in home-care services. Under these circumstances, health care providers require quick access to effective tools and health information resources to do their job effectively. It also extends to the communication between different health units to improve patient care services. For example, a patient can use such systems during an emergency to check bed availability in a local clinic before leaving home. Health Canada is currently supporting two PDA-related projects in Saskatchewan which are listed below [15]:

- The Personal Digital Assistant and Nursing Software Evaluation Project that aims to evaluate how technology can improve the quality of health care in the province.
- PDAs for Primary Care Nurses in Northern Saskatchewan First Nations that aims to use PDAs to support primary care nursing practice and better meet patients' healthcare needs.

These two healthcare systems have been identified to clearly improve nursing practice in the rural and isolated communities [15].

2.4.5 Diseases and Epidemic Outbreak Tracking

This category of healthcare systems focuses on collecting and transmitting information about incidences of communicable diseases thereby helping in the containment of such outbreaks. The information collected usually include the location and levels of such diseases that health care authorities can use to target medical

assets towards affected areas. It is particularly important in the developing countries where collecting localized medical data is tough because a lot of affected patients often don't have access to hospitals that can measure and collect their health information in order to help healthcare authorities develop and gauge medical policies. Many low-income countries are already making use of healthcare systems to survey their population's medical conditions [15].

2.4.6 Remote monitoring

This category of healthcare systems makes use of medical information exchanged via electronic mobile communications to monitor and improve patients' health condition usually with the help of medical sensors. Usually, remote monitoring systems read physiological information about chronic health conditions such as diabetes from the patient, and send them through mobile devices to a centralized repository hosted by healthcare providers for analysis. The information gathered can be used for the early detection of extreme critical conditions that can lead to loss of life. However, remote monitoring is usually very costly to maintain.

2.4.7 Remote data collection

This category of healthcare systems collect data for use by healthcare authorities and providers which can help in the development and improvement of policies especially in situations where information is needed in real-time thereby making paper-based reports inefficient. This is very important for conducting surveys in developing countries. Such systems involve the use of mobile devices by health workers to collect and enter health data after which the information is transmitted to a centralized database for analysis. Nevertheless, systems designed for remote data collections are usually very costly to implement.

2.5 Existing Healthcare Systems

In this section, we provide a review of existing healthcare systems for the elderly. One such system is *Codeblue* developed by the Harvard Sensor Network Lab which is heavily reliant on context awareness [68]. *Codeblue* architecture involves the use of multiple medical sensors for the tracking of patient's health information by end-user devices through wireless communication. Each patient has a channel that sensors can publish data to and end-users such as doctors and nurses can subscribe to through their hand-held devices. However, the *Codeblue* project has been known to be subject to numerous attacks such as denial of service, routing loop, and grey-hole attacks [56]. With the anticipation that *Codeblue* will be used in the future for emergency care and stroke rehabilitation of elderly patients, authors of the *Codeblue* projects will need to include adequate security measures in their design.

UbiMon is a context-aware healthcare architecture developed by the Imperial College, London that is aimed at capturing detectable and predictable life threatening abnormalities in elderly patients [78]. According to

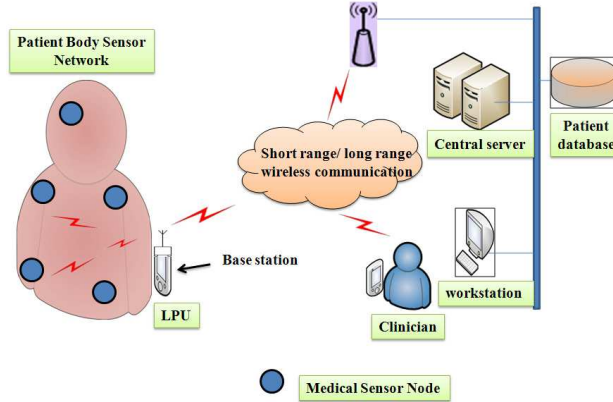


Figure 2.1: Ubimon System Architecture [62]

[62] and shown in Figure 2.1, the *UbiMon* architecture consists of nodes that are integrated with sensors, Local Processing Units (LPUs) that are used to detect health abnormalities from the information gathered from the nodes and a Central Server (CS) that analyzes the information and make healthcare recommendations to physicians. However, it is reported in [62] that the authors of *UbiMon* didn't meet up with the security requirements for wireless healthcare monitoring while implementing the system.

An architecture for monitoring the mental health of the elderly, especially those suffering from bipolar disorder (BP), has been proposed in [35]. This architecture consists of two levels, namely Personal Ambient Monitoring Infrastructure known as PAM-I and Personal Ambient Monitoring Programming Architecture known as PAM-A. PAM-I consists of medical sensors implanted into individual's body and environmental sensors placed in the home environment. The mobile phones are responsible for aggregating information from the sensors using rule-oriented applications and sending it to the personal computer for storage. The information measured by environmental sensors is transmitted to the personal computer through common communication protocols, for example Bluetooth [23]. The PAM-A module, on the other hand contains applications for handling inter-device network communication. This project is focused only on providing a reliable and acceptable solution for users thereby less emphasis was placed on patient's privacy.

Another healthcare system that exists for the elderly is *Alarm-Net* developed at the University of Virginia which is targeted at assisting elderly patient in their daily living [62]. *Alarm-Net* makes use of wearable sensors to measure individual physiological data and the data is transferred along with its source address, ID and sensor type in real-time. Network security is achieved in Alarm-Net through the use of secure remote password (SRP) protocol. Further, its sensors make use of AES encryption scheme to protect information [112]. According to [62], the major drawback of this approach of sensors using built-in cryptography is that there is no way to decrypt the information while in communication which can be beneficial in cases where it is necessary. In addition, there have been cases where confidentiality was compromised such as leakage of location [81].

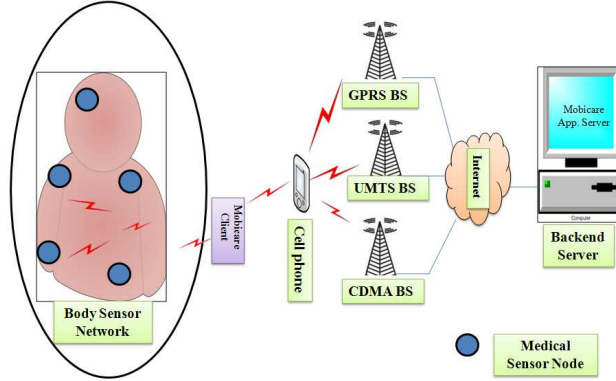


Figure 2.2: MobiCare System Architecture [62]

Another project that extensively makes use of context-awareness is *MobiCare* designed in 2006 funded by the European Commission [28]. *MobiCare* is used to monitor elderly patients over a wide-area. The main motivation behind the development of *MobiCare* is improving the quality of life for the elderly and in essence save lives. The architecture consists of a sensor network containing wearable sensors, a BSN manager and a back-end infrastructure known as the *MobiCare Server* [62]. The sensors are responsible for gathering information and sending it to the *MobiCare client* which makes use of application layer standard HTTP POST protocol to store the information in the *MobiCare Server* as shown in Figure 2.2.

A most recent architecture for elderly patient’s monitoring is *MEDiSN* developed by John Hopkins University[61]. It contains several battery powered physiological monitors and medical sensors that measure physiological health information. The monitors are responsible for temporarily storing health information, encrypting and signing it and then transmitting it to relay points (RPs). These relay points are implemented as bidirectional routing tree. *MEDiSN* makes use of a database to store health information and provide them to users through specialized GUI clients [61].

Another wearable sensor-based mobile healthcare system that reads context-based information through the use of small and battery-operated beacons such as motion and location of an elderly is *CarePredict* [2]. *CarePredict* transmits data through wireless communication service to remote servers from where experts can examine the information in order to detect any acute deviation, such as restless sleep patterns and changes in eating patterns of the elderly which can then be isolated and investigated further. *CarePredict* allows caregivers and family members to track information about their loved ones from any computer and subscribers can grant or revoke access at any time they want.

Other context-aware healthcare projects include those being developed by University of Rochester, Georgia Tech, Massachusetts Institute of Technology (MIT), TIAX, and LLC. All are “smart living” projects where context-aware and ubiquitous sensing, computer vision-based monitoring and acoustic tracking through the use of infrared sensors and video cameras in laboratories are combined in order to monitor health information

for long periods of time[113]. *Guardian Angel Service*, developed for the Symbian OS platforms, also provides active context-aware monitoring for medical stakeholders about elderly facing chronic conditions [70]. It uses context-aware sensors to read vital signs like heart rate and skin temperature in order to help the patient avoid hazardous health conditions. *CareMerge* is an enterprise mobile healthcare system that helps provide real time information about the health of the elderly to their healthcare providers and family [1]. It offers communication with and notification to family members, tracking and sharing of health information and automatic reminders. *Virtual Health Pet* system, developed in Brazil with Java Micro Edition (J2ME) technology [76], sends out alarm to remind the elderly on medication, as well as alerting emergency services. *GetMyRx* system attempts to simplify the administrative task behind the elderly people in getting their prescriptions [8]. It involves scanning and sending of paper prescriptions with name and address of the patient to a local pharmacies for delivery. Doctors are also able to send prescriptions directly to pharmacies from the comfort of their offices. *Chinese Aged Diabetic Assistant*, developed by Microsoft in China [3], is a smart-phone-based support system for elderly diabetics patients that provides recommendations and guidelines for patients in taking insulin and oral medications. *Mobile HIV/AIDS Support* [9] assists healthcare workers in rendering quality services in the developing world by providing reliable medical information for use while the health staff are in the field. *EpiSurveyor* [7] is an easy to use open source software developed by Washington-based non-profit software company *DataDyne* that helps in the creation and sharing of surveys for development of policies by healthcare authorities.

From the review, it is clear that the current systems are more focused on providing information to medical professionals and helping them to reduce administrative work than to providing direct healthcare support for diagnostic or clinical purposes. Moreover, these systems are not patient-centric, in the sense that they do not provide immediate feedback to patients nor do they put them in the healthcare loop. This is evident as most of the user interfaces for these systems are very complex thereby making interaction with such systems difficult for elderly patients. Some of the systems are also only targeted towards one specific health condition which makes it impossible for elderly patients who suffer multiple medical problems to use them. The architecture that is proposed in the thesis is based on “open-closed” principle. It is closed with respect to those who are registered with it and who are authenticated to use it. It is open in the sense that any one from anywhere and at any time can register into the system and get its services. Most importantly, the system is patient-centric. Thus, patients are part of the feed-back loop. They communicate directly with medical experts, get auto-feedback from the reasoners that adapt to their health contexts, and will have the ability to be part of “social network” to exchange their health history and treatment details within their choice of networked friends.

Chapter 3

Context Awareness In Healthcare

The health care domain is a rich domain for context-aware architectures that usually focus on monitoring and improving patient's health conditions over a period of time by processing and sharing EMR among medical stakeholders. EMR is a generic term that can be used to classify longitudinal electronic records of patient health information generated by one or more encounters in any care delivery setting and can be used in reducing medical errors as well as improving healthcare savings.

A category of electronic healthcare systems heavily dependent on context awareness are *mHealth* systems. *mHealth* can be defined as the integration of mobile computing, medical sensors and portable devices to ensure health care [55]. That is, properly configured medical devices, medical professional, and care givers when integrated with mobile computing, medical sensors and communications technologies [54] healthcare can be provided more efficiently with optimal cost. The *mHealth* domain arose out of the need to provide healthcare services to patients regardless of their physical location. Some other factors such as increased availability, better performance, context-awareness capabilities and cheaper mobile data rates have also contributed to this shift of interest in the *mHealth* domain. For example, it is reported in [60] that 83% of American adults have an active mobile phone while 35% of these users use their phones to access information over the internet. Another report [29] confirms that at the end of 2009, in advanced countries there were approximately 4.6 billion mobile cellular subscriptions, with the average penetration rate of above 100%. Individuals were within the reach of their mobile phones on an average of 58% of the time [82], thus making it easy for a system to monitor and understand personal habits and preferences. According to the report [29], in 2014 alone the use of some of the current *mHealth* systems have resulted in healthcare savings worldwide between 1.96 billion and 5.83 billion dollars. It is therefore no surprise that government institutions in industrialized countries as well as low and middle-income countries are continually indicating their intent in using mobile healthcare systems to achieve MDG through the development and deployment of nation-wide health services

made achievable especially in the areas of remote health-care monitoring of patients, maternal and child health as well as emergency medical care [88]. The information collected from these systems can further be used by health authorities for taking definite actions in improving the health conditions of backward regions.

However, most electronic healthcare architectures still face challenges in worldwide adoption due to their lack of robust security and privacy measures and inadequate knowledge integration to meet up with healthcare laws and regulations [79]. A national study [55] shows that there are a lot of security and privacy issues that must be integrated in the healthcare architecture before it can be readied for real deployment. This is because most of these architectures are designed to communicate over wireless networks thereby making them susceptible to common vulnerabilities. These vulnerabilities can be classified into two major categories, called *passive* and *active* [77]. A passive attack is usually a network attack where the attacker intends to gain information, while an active attack is one in which the attacker can change the communicated information itself. Thus in a passive attack *confidentiality* is compromised, and in active attack *integrity* is lost.

In this chapter, we give an overview of the concepts and terminologies of context-aware healthcare architectures. Section 3.1 provides an introduction to context-awareness. In Sections 3.2 and 3.3, we present the terminologies on which healthcare architectures are built on. In Section 3.4, we discuss the essential security and privacy features to integrate in the architecture.

3.1 Context

Context is very important to our day-to-day living as it provides relevant details about people, places, things and devices in the environment. It is how we interpret and interact with the environment. For example, we use context to sense and react to cases of impending danger. However, this innate ability does not automatically translate when we interact with computer systems. While it is easy for an health provider to interpret the current health condition of a patient, a computer system needs to be modeled in such a way that it makes use of certain tag values in making relevant and accurate decisions. For example, GPS coordinates may not bring relevant knowledge about the context of a dementia patient, but information such as elderly home or hospital are more useful.

Context can be defined as any information that can be used to characterize the situation of an event. A situation occurs when multiple entities around an event are assigned values. An entity is a person, place or object that is considered relevant to the interaction between a user and a system including location, time, activities and the preferences of each entity [51]. Another definition of context relating to ubiquitous computing regards it as any circumstance or condition surrounding a user that is considered relevant to the interaction between the user and the ubiquitous computing environment [90]. By extension, context-aware computing can be seen as the ability of a system to adapt to changing circumstances and respond according to the context of use [59]. A context-aware healthcare architecture intends to make use of the situations

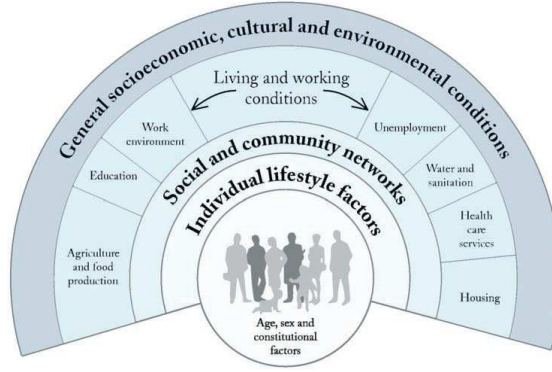


Figure 3.1: Health Determinants

around a patient to make better decisions with the overall aim of achieving the patient’s health goals.

Context awareness is currently being used in numerous other domains apart from healthcare. For example, in the domain of education, MOBIlearn is a project that aims at assisting individuals in learning by using context awareness to select relevant contents for such individuals [67]. Search engines are also known to use context awareness to provide relevant search results to users. Big companies are also attempting to use context-awareness to implement various context-aware solutions for example the Cool Town Project by HP as stated in [85].

3.2 Terminologies

This section discusses the terminologies used in the extension of our architecture. These terminologies were formed by adapting context awareness terms to the healthcare domain. By introducing these concepts, we hope to provide a solid foundation for understanding the architecture.

3.2.1 Health Determinants

By *Health Determinants*, we refer to factors that affect the health of an individual. These factors can be used to create a personalized health model for a patient, thereby assisting the physician in making decisions that will ultimately lead to the improvement of the patient’s health status. A patient’s health model can further be enhanced by including his desired privacy policies. Examples of *Health Determinants* include biological, environmental and physiological factors. Health determinants can be represented as either atomic or tuples or a collection of atoms and tuples where an atomic representation refers to a tag-value pair with specific data types and units. The values can be monitored initially, at specific intervals or even continuously especially when they are critical to the health of the patient.

Fig 3.1, an enrichment of the health model [48], has 12 categories of *Health Determinants*. Their informal

meanings are explained below.

- *Income and Social Status*: By income and social status, we refer to the aggregation of information about the wealth, lifestyles and occupations of patients. These information, either in isolation or when combined can be used to estimate the health status of an individual or a group of people. For example, as shown in Fig 3.2, there is substantial evidence that people higher up the social hierarchy and with higher income usually have longer life expectancy as a result of having access to quality healthcare services as well as qualified medical advising. This relationship between social hierarchy and health status is denoted by a term called *Social Inequalities in Health* as stated in [42].
- *Social Support Networks*: Social support networks refer to support from families, friends and communities. Social networks can effect the health status of a patient in many ways. A California study was able to establish that the more social contacts an individual, the less unlikely the person is to die early as those social contacts are sources of emotional reassurance.
- *Education*: It has also been established that health conditions improve with level of education because the level of education contributes to social status. People with higher levels of education are also able to understand and follow medical advising information.
- *Employment and Working Conditions*: When people are unemployed or work in unsafe environment, they tend to have poorer health conditions because a good job does not only provide a source of income but also personal satisfaction and opportunities to expand social connections. For those working in unsafe environment, it is very easy for them to suffer stress and bodily pains with increased chances of high blood pressure and heart disease.
- *Social Environments*: Social environments refer to the values and norms by a society. These values influence the health conditions of individuals and populations. For example, it can be expected that in societies where government and organizations regularly hold health awareness campaigns, people living in such societies will make healthier choices thereby resulting healthier conditions.
- *Physical Environments*: Physical environments refer to environmental factors such as climate change, pollution and indoor air quality. Research shows that people staying in areas experiencing exposures to pollution are usually at the risk of suffering chronic diseases and communicable diseases.
- *Personal Health Practices and Coping Skills*: Personal Health Practices and Coping Skills refer day-to-day decisions individuals make in order to maintain sound health as well the effect of social, economic, and environmental factors on these decisions. For example, smoking has been found to be responsible for a significant percentage of deaths for adults between the ages of 35 and 84.
- *Healthy Child Development*: Recent research reveals that experiences while at a young age have signifi-

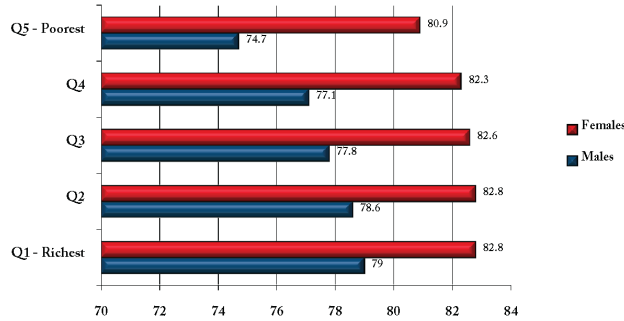


Figure 3.2: Life Expectancy In Years

cant influence on an individual’s health condition. According to the research study, positive stimulation in between birth and age 6 influences the connecting and sculpting of the brain’s neurons thereby improving learning, behavior and health. For example, a child who grows up in a dysfunctional home environment is more likely to experience increased emotional vulnerability later in life.

- *Biology and Genetic Endowment:* Biology and genetic endowment of an individual is a significant determinant of health. Genetic endowment in some situations determine how individuals respond to epidemics thereby determining the overall health condition of such individuals.
- *Gender:* Gender refers to the array of roles and personality traits attributed to the two sexes. As shown in Fig 3.2, men are more likely to die prematurely than women as a result of higher susceptibility to heart disease, fatal unintentional injuries, cancer and suicide. For example, rates of potential years of life lost before age 70 was found to be almost twice as high for men than women and approximately three times as high among men aged 20 to 34.
- *Culture:* Culture is a system of ideas and values an individual lives by every day. A persons cultural background has an important influence on his or her emotions, diet and attitudes to illness all of which can influence health and the use of health care [50].

3.2.2 Health Conditions

The term *Health Condition* refers to the medical diagnosis of a patient done by healthcare professionals. For example, based on the medical tests, a patient’s health condition may be determined as ‘depression‘ or ‘dementia‘. It is important to note that a patient can have multiple health conditions and can also be at risk of developing additional health conditions that can result from one or more of existing health conditions. We intend to categorize health conditions in terms of the health determinants that can be monitored to assess a patient’s health progress over a period of time. For the purpose of this thesis, we selected **dementia** and **hypertension** as our major case studies.

3.2.3 Health Situations

We use *Health Situations* to represent states of interest to the healthcare application in our architecture. Usually, for each health condition, there will be a certain set of health situations that are important for consideration. For example, the *High* health situation is realized when a patient diagnosed with a *diabetic* health condition records a blood sugar level of 10.5mmol or above. *Health Situations* can have relations between each other. For example, an *Emergency* situation usually follows a *Danger* situation. It is important to note that *Health Situations* are constructed by the medical experts, based on medical knowledge and clinical experience. As such, a health situation is domain-dependent and context-sensitive.

Health Situations are of more importance to our healthcare architecture than the observed health readings through sensors. For example, an *Emergency Situation* defined for a certain health context is more helpful for making decisions than the observed temperature value of 100°C. Therefore, it is important to have a component that can relate observed health readings to the corresponding situations and reason on predefined reaction choices. This module should be designed to take in observed values and perform data transformation in order to relate to corresponding *Health Situations*. In cases where a *Health Situation* depends on more than several determinants, this module also aggregates the different values to perform a look up of the corresponding situation.

3.2.3.1 Situation Expression Language (SEL)

In order to interpret situations syntactically, there is a need for an expression language as discussed in [18]. This expression language allows for mathematical representations of situations thereby making it easier for electronic systems to interpret and make adaptations based on such context. The following operations are supported in SEL:

- Comparison operations, for example *Equal* and *Not Equal*.
- Logical operations, for example *AND*, *OR* and *NOT*.
- Basic operations, for example *Addition*, *Subtraction* and *Division*.

The language can support expressing *Hybrid Situations*, which are situations dependent on other situations, as well as *Literal Situations* that are dependent on only their dimension values. This expression language was defined using a CFG.

Situations are defined in the expression language within curly brackets as shown below:

$$\langle \textit{Situation} \rangle ::= \langle \textit{SituationRule} \rangle \textit{ or } \langle \textit{LiteralExpression} \rangle \quad (3.1)$$

$$\langle ANDSituationRule \rangle ::= \langle SituationRule \rangle AND \langle SituationRule \rangle \quad (3.2)$$

$$\langle ORSituationRule \rangle ::= \langle Situation \rangle OR \langle SituationRule \rangle \quad (3.3)$$

$$\langle NOTSituationRule \rangle ::= NOT \langle SituationRule \rangle \quad (3.4)$$

$$\langle SituationToken \rangle ::= Identifier \quad (3.5)$$

$$\langle LiteralExpression \rangle ::= \{ \langle DimensionRule \rangle \} \quad (3.6)$$

3.2.4 Health Context (HT)

We use *Health Context* to represent the observations of continuous monitoring of a patient, done through the sensor network for each patient. For example, the ID, time, and location of a patient together with the blood pressure reading is a health context. It is possible to represent observed measurements differently based on the device used to measure. For example, for sugar level reading the units are different in Asia and in North America. While keeping track of *Health Context*, it is important to consider the parameter type and unit of the reading in order to ensure accuracy and compatibility.

3.2.5 Health Adaptation (HA)

Health Adaptation is the set of predefined execution plans, usually provided by healthcare experts, for each specific health category. Examples of health categories are *diabetics*, *hypertension*, *depression*, and *dementia*. Health situation for each health category identifies the significant levels that should be monitored for patients in that category. However, it can be specialized for a particular patient within each health category by taking into account the patient’s context history, personalized health model, and the other health categories to which a patient might belong. For example, patient A and patient B may both belong to *diabetics* category, however patient A might also have *hypertension* while patient B might have *depression*. So, even if their blood sugar level readings are identical, it may have to be interpreted differently for the two patients. Our architecture also considers past histories of health contexts to cater for cases where a particular context might need to occur for a number of times before its associated *Health Adaptation* can be triggered. An example of *Health Adaptation* is sending a message to emergency ambulance services to pick up a dementia patient who is wandering outside the healthcare facility while her health context (as monitored by a glucose meter embedded in her body) suggests her health situation to “high sugar level”. As suggested by this example, an adaptation might involve a sequence of actions, depending upon the message type and complexity. A simple message like “trace the patient” will have to be interpreted according to the healthcare policy in force at “emergency service department”. In order to express adaptations (and their associated actions) we need a rich language. The virtue of such a language, in which several types of actions such as “sequential actions”, “conditional actions” and “repetitive actions”, is that it can be coded in a high-level language for automation through existing actuators. As we explain in the following section, this language cannot be a

general purpose programming language, yet it will inherit the features of such a language.

3.2.5.1 Workflow And Policy Expression Language (WPL)

In this section, we discuss the language for expressing and executing adaptations. This expression language is defined using a CFG whose grammar is shown in the Appendix. In simple terms, using WPL a collection of workflows can be specified, where a workflow is defined below.

$$\langle Workflow \rangle ::= \langle StatementCollection \rangle \quad (3.7)$$

The *Statement Collection* can contain multiple statements or just one statement. for example, the *While* statement can be used to express repetition of a set of actions, subject to finite termination. For checking policies, a *Condition Statement* can be used as a part of an *IF Statement* or a *While Statement*.

3.3 Entity Model

In this section, we provide the entity model of our healthcare architecture. Figure 3.3 shows the general overview of the essential entities in the architecture. This model can be extended to integrate more entities, such as *hospitals*, *government healthcare agencies*, and *healthcare insurance providers*.

3.3.1 Elderly Home (EH)

We use the entity *Elderly Home (EH)* to represent a smart physical accommodation for elderly people where they are managed and monitored by primary caregivers through the use of wearable or implanted photoelectric small storage devices known as sensors for measuring and monitoring changes in environment. These sensors can either be categorized as hardware-based or software-based or hybrid of the two categories. The *range* of a sensor refers to the intervals that a sensor is designed to measure. To be more precise, the range of a sensor represents its highest and lowest measurable values for example a given pressure sensor may have a range of -400 to +400mmHg. The accuracy of a sensor refers to how far a sensor can deviate from the ideal value it is meant to measure, for example a temperature sensor accurate to 0.0010°C is expected to agree within 0.001°C. The sensitivity of a sensor refers to the least input value to cause an output change in such a sensor which in essence the slope of the output characteristic curve. An example is a blood pressure transducer with a sensitivity of 10 mV/V/mm Hg that will output 10-mV output voltage for each volt of excitation potential and each mmHg of applied pressure.

A sensor can either monitor changes either in real time or at specific intervals. For sensors reading data in real time, it is very important to be able to process the data as it is measured with minimal errors. Examples of sensors commonly used in the healthcare sensors include blood glucose, blood pressure, gyroscope, and pulse oximetry sensors. . These sensors are usually connected in a star topology through wires and short-

range wireless techniques such as IEEE 802.15.1/Bluetooth or IEEE 802.15.4/ZigBee, or a combination of the two [89]. For example, a patient suffering from diabetes may use two different sensors, one measuring the glucose level and the other measuring body mass index (BMI). Another example is the detection of location from a GPS sensor or user's IP address where many readings that are available and just one value must be selected from these readings. Each sensor in the elderly home acquires data, converts it into an electrical signal and amplifies it for communication with the module responsible for validation and aggregation. This aggregator module can also request readings from a particular sensor at a specific point in time. In order to support this, sensors usually cache latest readings of parameters with the appropriate expiration dates. This architecture is popularly known as BAN.

Electronic communication in the Elderly Home is often done through a mobile terminal which is simply a device such as a phone or personal display assistant. They have the ability to send information over a network. Usually, the mobile terminal has an application running on its operating system that acts as a hub that records all the data coming in from all the sensors and actuators, stores temporary information in a local database and then sends data out of the BAN to the Cloud Cyberspace through Inter-Ban communication. Most mobile terminals run one of the following operating systems: Android, IOS and Windows Phone.

3.3.1.1 Android

Android operating system is an open source operating system that runs on touchscreen mobile devices. It is based on the Linux kernel written in C and was developed by the Open Handset Alliance led by Google. It was first launched on November 5, 2007 and since then has proceeded to become one of the famous operating systems in the world having released 9 major versions: Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat and Lollipop. Applications that run on the Android operating system are usually developed with the Java programming language and can be deployed through the Google or Amazon playstore.

3.3.1.2 IOS

IOS operating System is an operating system developed by Apple Incorporation. It was released in 2007. Since then, IOS has evolved from version 1.0 to version 6.0. Applications running on the IOS platform are usually developed with the Objective C programming language and can be deployed through the Apple Store. IOS boasts of encrypting all personal information as long as the user uses a passcode.

3.3.1.3 Windows Phone

Windows Phone operating system is an operating system developed by Microsoft. It was first released as Windows Mobile in 2003 and was the most popular operating system in the United States at some point. However, due to the wide acceptance of the IOS and Android operating systems, the Windows Mobile has

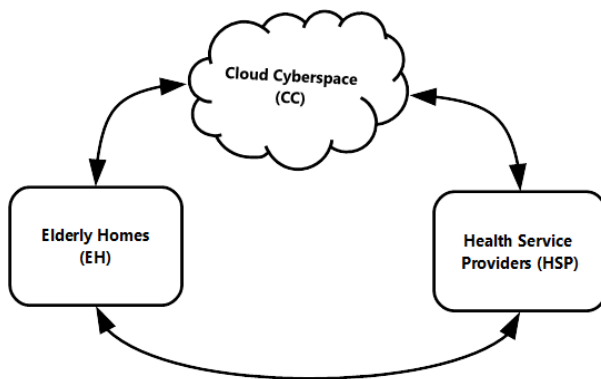


Figure 3.3: Entity Model

lost most of its popularity. On realizing this, Microsoft announced a similar operating system called Windows Phone. Applications running on the Windows Phone platform are usually developed with the Visual C++ programming language and can be deployed through the Windows App Store.

3.3.2 Inter-BAN Communication

Inter-BAN communication makes use of wireless network technologies such as 3G and WLAN. The connection usually involves the transfer of data outside of the Body Area Network to a computer server over HTTP.

3.3.2.1 HTTP

The HTTP is a stateless protocol in the application layer for the transmission of information in the client-server model. Each session involves connection from the client to the server through TCP including certain request methods indicating the action to be performed on a resource and the server responding back to the client with the outcome of the request. Usually this connection is susceptible to eavesdropping and man-in-the-middle attacks which has led to the development and use of SSL for the encryption of HTTP connections.

3.3.2.2 3G

3G is the third generation of wireless broadband technology and supports services that provide an information transfer rate of at least 200 kbit/s. It is based on a set of standards used for mobile telecommunications by the International Telecommunication Union. 3G has opened a new variety of applications especially in the mobile health sector.

3.3.2.3 WLAN

The WLAN is a wireless technology for two or more devices that uses high-frequency radio waves and often include an access point for the transmission and reception of radio signals to the internet. Usually, WLANs

are compromised by using specialized software to break algorithms used by the access point for the encryption of information transmission. Typically in WLAN, users can transfer data for up to 54Mb/s with a distance of 500m.

3.3.3 Cloud Cyberspace (CC)

The *Cloud Cyberspace (CC)* represents the server where all incoming medical information from the elderly home is received, persisted and made available to healthcare service providers. The infrastructure of the system is based on a concept called known as *Cloud Computing*. The term *Cloud Computing* actually became popular in October 2007 when IBM and Google combined to form a collaboration in the area of interoperability [17] with the first major implementation launched in November the same year by IBM [65]. After this, multiple similar projects have implemented by various technology companies such as Yahoo and Intel Corporation [109]. According to [17], *Cloud Computing* implies a *SOA* that offers reduced information technology overhead for the end-user as well as great flexibility, reduced total cost of ownership and on-demand services. The *SOA* involves data requests to an end-point by end-users at desired functional, quality and capacity levels and reception either at the time requested or at a specified later time [17]. The first implementations of the *SOA* architecture included the remote procedure calls (RPC), DCOM and Object Request Brokers (ORBs) based on the CORBA specifications [5].

As indicated in [66], *Cloud computing* provides services to users in three major layers discussed below:

- IaaS: In IaaS, the cloud provider hosts the service infrastructure which includes operating systems, storage and deployed applications on behalf of its users and provides a set of virtualised service interfaces through which users can manage these resources remotely. The provider also assists in performing certain tasks such as maintenance and backup. Compared to the users at other layers of cloud computing services, an IaaS user has more access and control over software components and usually pays for the services on a per-use model basis.
- PaaS: In PaaS, the cloud provider provides an environment for developers to implement their applications and services over the internet. These services can then be accessed through the use of a web browser. A consumer of PaaS is usually able to manager or control the applications and some of the hosting environment settings but usually can't access the underlying platform like a IaaS. PaaS services are usually paid for on a subscription basis usually per year.
- SaaS: In SaaS, the end users access software applications over the internet without having to manage or control the underlying cloud infrastructure thereby removing the need for investment in additional hardware. A good example is web-based email. SaaS users usually subscribe on a monthly basis.

A comparison of the summary of differences among the three service layers is shown in Table 3.1. Through

Type	Consumer	Service Provider	Service Level Coverage	Customization
SaaS	End User	Finished Application	Application uptime and performance	Minimal to no customization, capabilities dictated by market or provider
PaaS	Application owner	Runtime environment for application code, cloud storage	Environment availability and performance, no application coverage	High degree of application level customization
IaaS	Application owner or Middleware and application support	virtual server, cloud storage	Virtual server availability, time to provision, no platform or application coverage	Minimal constraints on applications installed on standardized virtual OS builds

Table 3.1: Comparison of cloud service models [4]

the power that *Cloud Computing* offers, patient’s personal health information is shared among multiple health providers by the Cloud Cyberspace (CC). The Cloud Cyberspace makes use of the *Workflow Policy Expression Language* to analyze medical information for predefined states and executes the corresponding response as stated by the healthcare experts.

The Cloud Cyberspace is expected to share patient’s personal health information with multiple clients and usually runs a web server such as Apache and IIS which is used for processing and persisting data usually in a database like MySQL or SQL Server.

3.3.3.1 Apache Web Server

The Apache Web Server is a free open-source web server launched in 1995 and widely used by server administrators worldwide. It is available for a lot of operating systems including linux and windows. Apache usually runs under the process name “httpd”. The latest version for Apache is 2.4.12.

3.3.3.2 Internet Information Services(IIS) Web Server

The IIS web server is a web server developed for Windows NT by Microsoft for use with Microsoft Web Technologies such as the ASP.NET, Windows Communication Foundation (WCF) and Sharepoint applications. The latest version for IIS is 8.5.

3.3.3.3 MySQL

MySQL is an open-source relational database management software currently managed by Oracle Corporation. Many web applications such as Wordpress and Drupal rely on MySQL as their primary datastore due to its low management costs and enterprise capabilities. Although MySQL is widely accepted and used, it is not fully compliant with functionality standards for relational database management softwares. The latest MySQL version is 5.7.

3.3.3.4 SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft for enterprise use. It runs Transact-SQL and provides enhanced reporting, scalability and security. The latest version is SQL Server 2014.

3.3.4 Service Providers (SP)

The *Service Providers (SP)* represents the generic actuators in the system that provide healthcare services to patients. These actuators can also either be hardware-based, software-based or even human-based. Hardware-based actuators usually work by converting energy into motion. Actuators work by subscribing to events and getting instructions included in the event for example an emergency service can subscribe to be notified when an elderly patient requires such services [92]. It is possible to have multiple actuators interacting with a context-aware architecture through the use of public interfaces.

We provide a summary of the actuator types that we expect to interact with our architecture below:

- *Caregivers*: Caregivers provide paid or unpaid daily care to the patients. The usual duties that are expected from caregivers include managing medications, providing personal hygiene and interacting with doctors and nurses on the behalf of the patient. It is assumed that each caregiver owns a mobile device through which they are able to subscribe for notifications from the *Cloud Cyberspace (CC)* based on the current context of the patient. We also expect that a patient can be linked to multiple care givers. For a diabetic elderly patient, a caregiver might be required to administer insulin at certain specific times everyday. ited
- *Doctors*: Doctors provide services to patients that assist in the diagnosis and treatment of diseases. Before a patient can be admitted into the *Elderly Home (EH)*, it is expected that he or she would have been diagnosed with a particular medical condition and an appropriate treatment plan would have been recommended. Doctors can subscribe to the *Cloud Cyberspace (CC)* to receive notifications when situations that require a change in the treatment plan of a patient is detected.
- *Pharmacists*: Pharmacists provide services to patients that ensure safe and effective medication use. Pharmacists can subscribe to the *Cloud Cyberspace (CC)* to request for drug dispatch requests based on the treatment plan created by the doctor.
- *Emergency Response Officers*: Emergency Response Officers are health service providers that are trained to be the first line of response in any emergency situation. Usually, their job include providing an initial assessment of the situation and then picking up the patient for emergency admission in the hospital. Emergency Response Officers can subscribe to the *Cloud Cyberspace (CC)* for notification on

critical situations.

3.4 Context-dependent Security and Privacy

For us to have a dependable healthcare architecture, it is important to consider the serious challenge of protecting the information stored on these systems (security) and ensuring the information is used for appropriate or specified purposes (privacy). Most healthcare systems consist of wearable or implantable medical sensors that measure physiological information about patients and send them to a remote server through electronic devices. This architecture opens door to multiple entry points for attackers to compromise the system ranging from threats that focus on eavesdropping on information while in transit to threats that targeted towards making the system unavailable for usage. Over 18 million patients had their health records breached between 2009 and 2011 with an average of 49,000 records are breached per incident, thereby making the impact of breaches significant [96]. However, a report identified that there are more data breaches due to loss or theft than hacking [22].

The consequences of information breach on context-aware healthcare architectures are overwhelming. Healthcare providers can lose data, revenue, trust, customers and opportunities as a result of having their system compromised. In fact, for healthcare providers to benefit from the 2009 American Recovery and Reinvestment Act (ARRA), they must ensure that the health information collected from patients are stored with appropriate security and privacy measures and also meaningfully used [98]. A stolen personal health record is estimated to worth \$50 on the black market which is much more than the cost of a stolen credit card with CCV estimated to worth between \$1 and \$6 [96]. It was further pointed that a US health insurance firm recently incurred fines and penalties worth \$250,000 as a result of a recent breach of 1.5 million medical records. Apart from the financial consequences, many users usually lose faith in healthcare systems once there has been at least one case of breach reported. *It is therefore important to consider the security and privacy of patient while developing healthcare architectures.*

3.4.1 Common Issues

Surveys revealed that a lot of users of healthcare systems are unaware that they are targets of cyber-attacks and what they can do to mitigate such threats. For example, as at 2012, 44% of adult mobile users are not aware of the vulnerabilities and solutions for their mobile systems [99]. According to the report, the number grew substantially to 57% in 2013. Usually, this lack of awareness is enhanced by the fact that hackers can easily cover their tracks in the mobile ecosystem as mobile devices have restricted user interfaces and low resources especially in the mHealth domain as most of such systems usually collect large amounts of sensitive personalized information about subscribers and this can lead to adverse consequences if such information is made available to unauthorized or unavailable to authorized users. An unauthorized user can range from an

attacker that is attempting to view information being sent over a public network to a third party accessing information from the mobile device if it is lost by the user.

However, most existing mHealth systems have not incorporated security. According to a recent survey [20], only 38% of mobile health care systems have clearly defined mobile privacy policies that they follow in ensuring patient's privacy. Another study of mobile health and fitness applications reported that only 25% of free mHealth applications and 48% of paid mHealth applications informed their users about the existence of such policies [71]. Some of these applications also share personalized health information with third party systems without informing users [49].

In addition, because most mHealth systems make use of sensors to transmit sensitive detailed health information about patients over a wireless or wired network, an attacker can install a malicious node within the range of the network and intercept information. This activity is referred to as passive eavesdropping. The next higher level of attack is the activity where by using a malicious node to grab information, via sending queries to an existing transmitter, is referred to as active eavesdropping. Such information may contain additional information about patients' habits, location and movements and therefore unintended disclosure of such information pose huge risk to patients for example an attacker can use the compromised message to physically harm the patient [47].

Most mHealth systems store information about patients on cloud servers susceptible to various common internet attacks [115]. The recent attack on HealthCare.gov website was due to the fact that the appropriate security measures for changing manufacturer's default password were not implemented on the cloud servers. In 2014, it is reported [21] that attackers targeted healthcare accounts more frequently than the business sector.

Another important security and privacy issue for mHealth systems is the situation where patients lose their mobile devices without implementing appropriate authentication mechanisms such as passwords on such devices [57]. This also applies to healthcare providers implementing the BYOD paradigm where employees bring their own personal devices for use at work. 2% of all mobile users in the UK reported theft in 2009 [47]. In addition, security issues could also occur in cases of ownership changes where the application stores information locally and the user forgets to wipe his health information before decommissioning.

Usually, most of the security and privacy issues existing in mHealth systems occur because no standard development procedures exist [57]. This implies that security and privacy issues are largely left to the manufacturers of mobile applications. Thus, not only the current apps are insecure, but also they are not *interoperable*. Consequently, the two basic principles of scalability and integration that are insisted for a disciplined development of software systems are lacking in mobile apps sector. Even when two apps may be interoperable the absence of secure protocols and varying degrees of security holes in the apps virtually

hinder scalability and integration. As an example of the severity, consider an elderly person who wants to pay for healthcare services while connected to a healthcare network. He will have to use many different apps, one for interacting with the healthcare facility, and another for mobile banking [83]. Yet, there is no way of informing the healthcare facility that the automatic transfer through mobile banking is meant for a very specific health service provided in the healthcare network. The lack of a standardized process also has implication in the appropriateness of security and privacy measures implemented in the mobile app. The app development team will be different from health professionals and consequently they lack the necessary domain knowledge that will drive security and privacy policies. Unintended data exposure can lead to loss of trust among clinicians and patients, especially in cases of applications that demand services from many domains.

3.4.2 The IAS-octave Model

In this section, we discuss an important security model known as the IAS-octave Model proposed by the Information Assurance and Security in 2013. This attempts to classify all the requirements for implementing software security. This model is basically an extension of the CIA triad which has existed as the model for information system security for a long time.

- Confidentiality: Confidentiality is keeping the information private from unauthorized users. It can also be seen as the right of an individual to have personal information kept private. If this right is violated, it could lead to grave consequences. Generally, confidentiality is achieved through cryptography.
- Integrity: Integrity in information security refers to keeping data as accurate and consistent as possible during its lifecycle [24]. In other words, information about users can only be modified by authorized parties especially in the healthcare industry where the personal health information about the patient is expected to be recent and only susceptible to changes by the patient himself.
- Availability: A secure system is expected to be available to users at all times. Information is unavailable not only when it is lost or destroyed, but also when access to the information is denied or delayed (usually referred to *Denial Of Service* attacks which is very common nowadays). In healthcare, availability is a very important property. A typical scenario is patient monitoring where the system is expected to provide real time updates to health care providers. If information becomes unavailable in such a system, it could lead to disastrous outcomes.
- Accountability: Accountability is being able to trace the actions of an entity uniquely to that entity. For example, the system should be able to link data violations to responsible individuals and impose appropriate corresponding punishments. In essence, accountability is answerability with sanctions.
- Auditability: Auditability is the capacity of a system to follow all activities associated with an infor-

mation resource. Usually, it involves a backward trace of activities to ensure accountability. Usually auditability is achieved through the use of audit logs or trails.

- Authenticity: Authenticity refers to genuinity of information and users working with the information. In other words, authenticity involves all parties involved provide a form of identity for authentication and authorization. There are three ways of authenticating a user which are things the user knows (for example, username and password combination), things the user has (for example, ID card) and things the user is or does (for example, fingerprints). Authorization is usually achieved through role-based access control.
- Non-repudiation: Non-repudiation is a legal concept which translates to the assurance that all parties involved in a digital transaction cannot deny their involvements. For example, non-repudiation can be used to guarantee that a sender of an email cannot deny having sent the email and the recipient cannot deny receiving the email.

Chapter 4

Architectural Design

In this chapter, the architectural design based on the entity model presented in Chapter 3 is presented. The chapter starts with a review of an existing context-aware architecture proposed in [108] while discussing its merits and inadequacies. Following that we propose a context-aware architecture that is more generic and expressive than the existing context-aware healthcare architectures.

The term **Architecture** refers to the combination of various components with the goal of achieving a specific goal. In essence, it is the blue-print engineers can use in implementing functional systems. A component based approach to developing context-aware systems offers many advantages such as reuse and reduced development time. A survey by [75] shows through an extensive use of case studies that trustworthy systems can be effectively modeled using the Component-based development methodology (CBD). From the survey, it can be deduced that this approach effectively contain complexity that exists in large systems. By following the approach, we were able to achieve results during the design of our architecture.

4.1 Review Of Existing Context-Aware Architectures

In [108], a three-tiered model was presented to manage the complexity in context-aware systems. The responsibility of each component and the way information is being passed around in the architecture is shown in Figure 4.1. The architecture handles perceiving, modifying and adapting to context. Perceiving context involves observing objects and their interactions with the environment which can be conveyed directly by the object or measured automatically by certain devices. This method of perceiving context was not properly explained in the architecture. In addition, the interface specifications with which the each Tier in the architecture communicate with each other were not properly specified which makes it look like a tightly-coupled system.

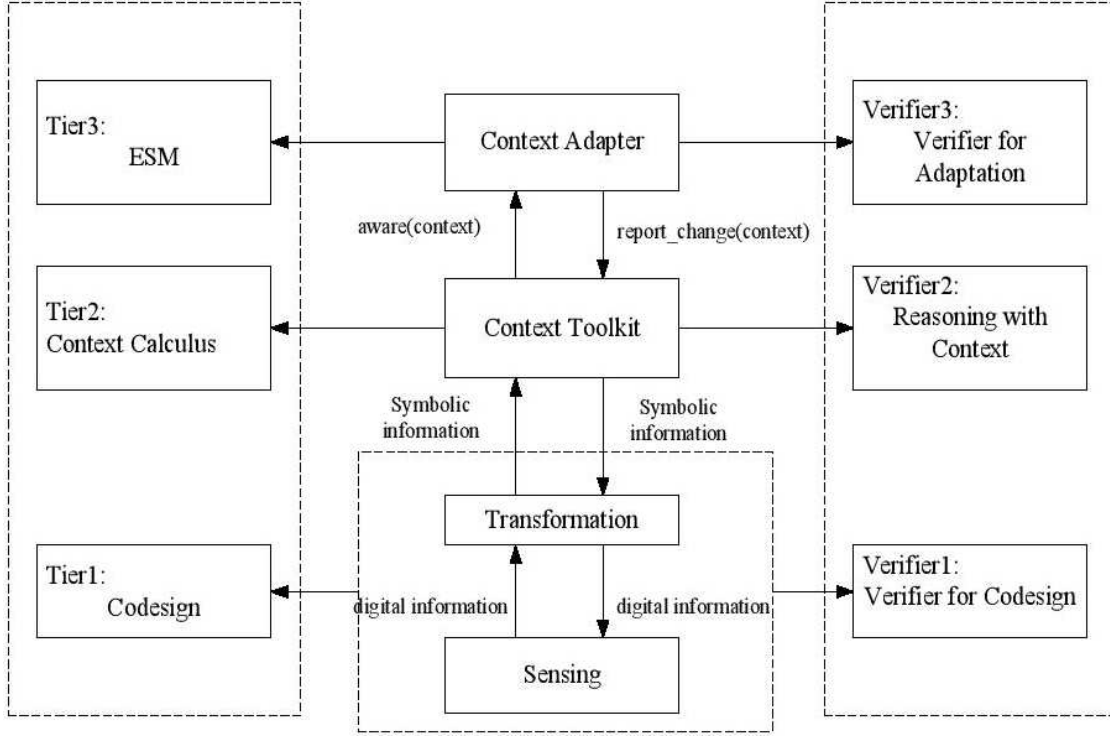


Figure 4.1: The Three-Tiered Formalism of Context-Aware System [108]

The information flow in the architecture involves Tier 1 notifying Tier 2 of any noticeable change in observations, Tier 2 performing deconstruction and modification of context through the use of context calculus and Tier 3 using the context to determine adaptations. However, the details about how adaptations are generated and executed were not properly discussed despite the case study of an *Anti-lock Braking System* that was presented in the study.

In summary, the architecture has several limitations including the following:

- the architecture is monolithic and therefore cannot be used for distributed systems such as large scale healthcare networks.
- the architecture does not give room for the inclusion of the personal healthcare model.
- the interface specifications through which each module in architecture communicates with another are not specified which makes extensions difficult.
- the architecture is more useful for embedded systems as it has no data store where health context, situations and adaptations can be stored and utilized.

Because of these limitations, there is a need for a better architecture.

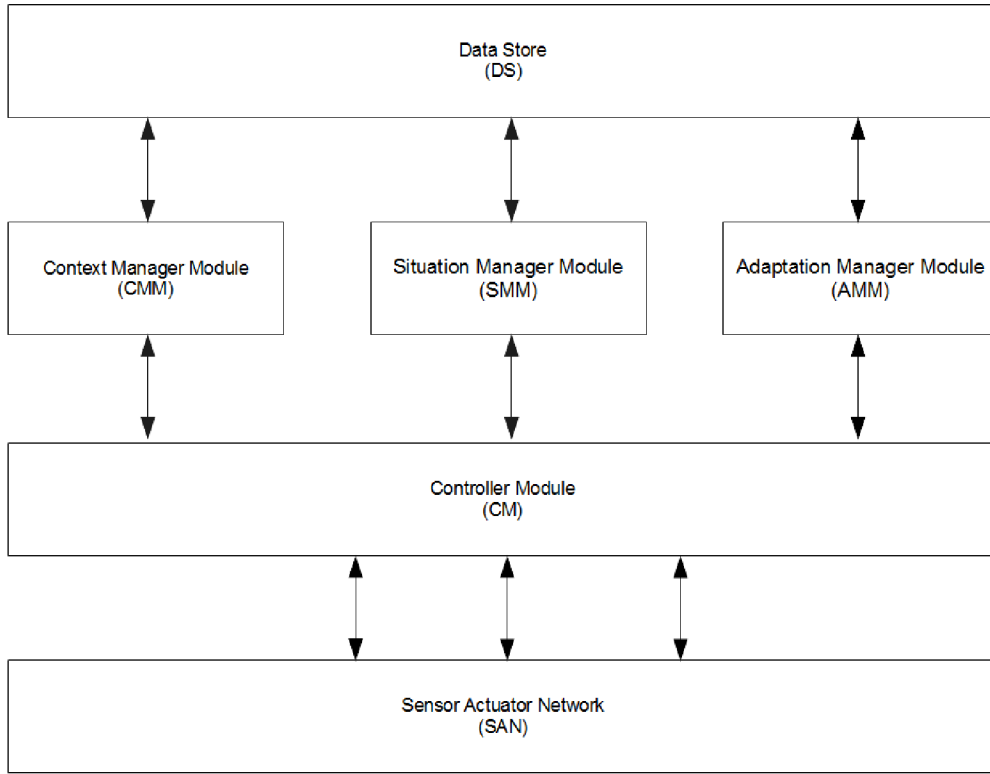


Figure 4.2: CAHA

4.2 Proposed Architecture

The architecture we propose in this thesis is based on *Component-Based Software Engineering (CBSE)*. An important virtue of CBSE is that it is possible to unify the specification of “heterogeneous” entities, such as medical devices, sensors, human (behaviors), actuators, and software artifacts. It is a bottom-up approach in which components are constructed individually with emphasis on interface specifications, individual components are tested based on the “provided services” providing them and policies for at the interfaces, and components are assembled using connectors to produce mini systems. Thus, the validity of each component and each subsystem can be established while the system is being built. In addition, the proposed architecture can be used on multiple platforms and can easily be ported to new technologies.

Figure 4.2 shows the architectural design for meeting healthcare needs of the elderly. The architecture contains five major modules: Sensor Actuator Network (SAN), Controller Module (CM), Context Manager Module (CMM), Situation Manager Module (SMM) and Adaptation Manager Module (AMM). A module is in essence represents a set of individual components designed to work with each other to achieve a specific purpose. Each module is loosely coupled and interacts with other modules through interfaces.

4.2.1 Architecture Overview

As stated previously, the architecture consists of six major modules. The Sensor Actuator Network (SAN) serves as the entry and exit points for the architecture through which external entities interact with the system. The Controller Module (CM) is responsible for routing data and commands among all the other components. The Context Manager Module (CMM) is responsible for validating parameter values and translating them into context values. The Situation Manager Module (SMM) is responsible for fetching the situations associated with the context and patient under consideration. The Adaptation Manager Module (AMM) is responsible for fetching and executing adaptations relating to the situations derived. For example, the adaptation for the context of a dementia patient wandering outside the elderly home could be to notify emergency services to pick up the patient. The Data Store (DS) is the repository where the personal health models of patients are stored. A more detailed explanation of each of the component is provided in the sections below. The main reason behind the separation of the modules is to ensure that each component can perform its functions independently and also making it easier to perform testing.

4.2.2 Data Store Module

The Data Store Module (DS) is responsible for storing the personal health models of patients. A unique identifier is assigned to each patient in order to associate the patient with his or her personal health model when the health model is being developed. In addition, the associations between sensors, actuators and patients are stored in the Data Store Module. The Data Store also keeps track of the association between sensors and the associated dimension types and units.

The Data Store Module stores each health model as a set of health conditions, health situations, health adaptations and privacy policies. Health conditions are linked to the patient in a one-to-many relationship. In other words, a patient is permitted to have one or more health conditions. Health situations are linked to each health condition in a many-to-many relationship as more than one or more conditions may be required to trigger a particular health situation. Also, a health condition will usually have multiple health situations of interest. Health adaptations or reactions are linked with health situations in a one-to-many relationship. Privacy policies are stored in a fashion where each actuator is linked to health determinants in a one-to-many relationship.

4.2.3 Sensor Actuator Network (SAN)

The Sensor Actuator Module (SAN) is responsible for measuring and responding to stimulus. Stimulus usually occurs when there is a change in environment parameters or when an external event occurs. For example, a stimulus event is created when an elderly patient suffering from dementia steps out of the elderly home. Also, when a patient suffering from hypertension has a blood pressure with systolic value more than

140mmHg or a diastolic value more than 90mmHg. Stimulus is usually associated with data parameters that can be measured by sensors through the use of *listeners* which are simply electronic components that can detect when an event occurs and measure the related dimensions. Based on our conceptual model explained in 3.3, communication between sensors and other components requires a channel using protocols such as TCP or HTTP as the sensors exist in the elderly home and the other components exist in the cloud cyberspace.

In our architecture, actuators also belong to this module. In general, an actuator is a device to convert an electrical control signal to a physical action, and constitutes the mechanism by which an agent acts upon the physical environment. An example is an insulin pump that can receive an instruction to either increase or decrease its rate of flow. In some other cases, actuators could also be software-based. However, our architecture focuses on human actuators denoted as **Service Providers** in our conceptual model. For example, caregivers, doctors and pharmacists that are tasked with interacting directly with the patients. Similarly to the sensors, these actuators receive instructions from other components over network protocols like TCP or HTTP. Our architecture is designed to use SSL to prevent eavesdropping of sensitive data while in transit.

4.2.4 Controller Module (CM)

The Controller Module (CM) is the *brain* of the architecture. It is responsible for managing the interactions between SAN and other modules. It is also managing the life cycles of other modules. In other words, the Controller Module is a container for other modules. This is very important in order that the different modules are independent of each other thereby resulting in a loosely-coupled architecture that promotes efficient testing and verification. This module also has knowledge of the interfaces exposed by other modules and how to call the actions on those interfaces with the data they require. For example, in order to validate a blood pressure reading, the Controller Module knows how to interact with the Context Manager Module in order to ensure that the reading is within the acceptable range of values.

The Controller Module is also responsible for translating message sent from and to the Sensor Actuator Network. For this to occur, the Controller Module has knowledge of the different data types and units that can be understood by each sensor and actuator in the Sensor Actuator Module as well as the input and output format used by the other modules in the architecture. In some other cases, the Controller Module works as an *synchronizer* in cases where the different sensor sources are sending similar tag values. For example, location can either be derived from both GPS and IP and therefore the Controller Module is responsible for choosing the value that is most relevant for the current context.

4.2.5 Context Manager Module

The Context Manager Module (CMM) is responsible for the generation and validation of health contexts. It receives the observed values from sensors, based on data policies it validates the information received from the sensors, and aggregates them to a context. In other words, the Context Manager Module is responsible for filtering out values outside the predefined input ranges of sensors. For each tag value, there is an expected range of values that can be regarded as valid which is known as the input range for that parameter. For example, the input range of a temperature sensor could be between -30 and 100 degrees Celsius. The Context Manager Module also intelligently takes into consideration the unit associated with the parameter. For example, when validating temperature values, a negative value is permitted for readings in Celsius while it is not possible to have a negative value on the Kelvin scale. In addition, the Context Manager Module ensures that the unit sent by the sensor is consistent with the unit expected for that sensor.

The Context Manager Module is also responsible for preventing replay attacks by checking the timestamps of readings sent from the Sensor Actuator Module. A replay attack happens when an adversary copies a request and tries to resend them at a later time. For example, an adversary can intercept packets of an authentication request and then, use it at a later time to fool the system into granting access. In order to prevent this in our architecture, we only consider readings sent within the last 15 minutes by comparing the timestamps sent with each reading. The timestamps are required to be in UTC format in order to ensure consistency. The Context Manager Module also validates that the reading originates from a registered sensor by comparing the identifier sent by the sensor with the table of sensor identifiers and any discrepancies are dropped.

After the Context Manager completes the validation of the sensor readings, it does a look-up of the patient associated with the sensor based on the unique identifier of that sensor. The architecture is developed under the assumption that there can only be one-to-one relationships between sensors and patients. That is, patients are not expected to share sensors or interact with sensors associated with other patients. Once the association is done, the Context Manager Module creates a new data format from the sensor readings which we refer to as *Context* which is then returned to the Controller Module.

4.2.6 Situation Manager Module

The Situation Manager Module (SMM) is responsible for resolving health situations based on the current context and the health condition for a particular patient. For example, for an hypertensive patient, the Situation Manager Module is responsible for detecting that a “Stage 2” situation exists when the context reflects that the patient currently has a systolic blood pressure of 160mmHg or above or a diastolic pressure of 100mmHg or higher. The Situation Manager Module is able to achieve it by using the identifier for the patient to find the related conditions and then do a look-up of situations that correspond to the current

context and the conditions found for the patient.

When a context value is received from the Controller Module, the Situation Manager Module fetches all the health situations that can exist for the patient under consideration. This is achieved by doing a look up of the identifier for the patient against the health situations table. The Situation Manager Module then searches through all the possible health situations for the patient in order to find the situations corresponding to the current context. In order to retrieve situations in an efficient manner, the Situation Manager Module takes into consideration the unit of the context determinant. For example, different algorithms will be required in order to fetch health situations when blood sugar determinant is measured in mmol/L or mg/dL formats. The situations are then returned back to the Controller Module.

4.2.7 Adaptation Manager Module

The Adaptation Manager Module (AMM) is responsible for generating the appropriate health adaptations or health reactions based on the general health situations existing for the patient. For example, the adaptation for an hypertensive patient experiencing a “Stage 2” context may be “Contact Doctor”. These adaptations are usually provided by medical experts and can either be specific to a particular patient or generic to all patients.

The Adaptation Manager Module is also responsible for evaluating the privacy policies in order to retrieve how much data each actuator has access to. These privacy policies are created by the patient when his personal health model is being assembled and added to the system. In order to create these privacy policies, a patient must select actuators and grant them access to only have access to certain parts of his record. This module is also responsible for logging adaptations as feedback in order to improve the personalized health model for the patient later.

4.2.8 Significance Of Architecture

The architecture is an expansion of the entity model discussed in 3.3. The Sensor Actuator Network in the architecture groups together the Elderly Home and the Service Providers in the entity model. The other major components in the architecture are implemented in the Cloud Cyberspace component described in the entity model. The Controller Module is a *Software Router* for processing requests and sending messages to the other modules except the Data Store. The Context Manager Module, Situation Manager Module and Adaptation Manager Module are *Component Models* that can be implemented in any programming language. The Data Store is a *Repository* of data objects that can be implemented as flat or database files.

The significant features of this architecture include

- **Modularity:** The components of the architecture are abstracted so that they can be easily imple-

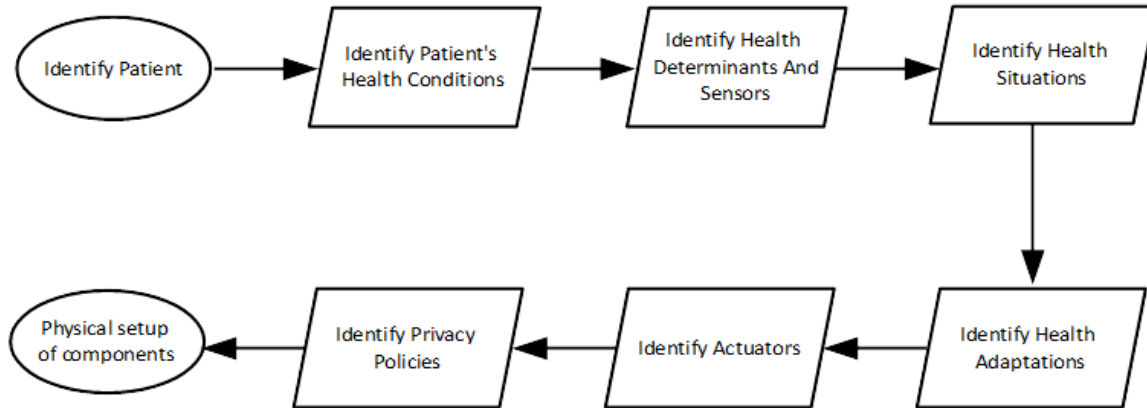


Figure 4.3: Process Flowchart

mented and maintained in any programming language of choice. Because each module is self contained, it can be replaced without necessarily affecting the rest of the system.

- **Separation of concerns:** The components of the architecture are developed with as little overlap in functionality as possible. More exactly, each component in the architecture is responsible for just one functionality which it does well.
- **Interface driven:** Each component has a clear contract by which other components can interact with it. The contract describes the preconditions, post conditions and the exceptions of the component. This is advantageous as information can be passed through components without actually having information about the underlying implementation of the different component.

4.3 Process Flow

By process flow, we refer the steps to follow while setting up a new patient on the architecture. As shown in Figure 4.3, the steps are highlighted below:

1. We assume that there is a patient X suffering experiencing one or more health conditions. Examples of possible health conditions include diabetes, hypertension, dementia, depression.
2. The PHM of patient X is then constructed by putting together the health determinants that are of interest to patient X's health conditions. For example, if patient X is hypertensive, patient X will require constant blood pressure monitoring. On the other hand, if patient X is suffering from dementia, patient X will require constant location monitoring in order to prevent wandering. Based on the health determinants that are of interest to the patient X's health conditions, sensor devices that can measure these determinants are selected and linked to the patient.

3. The situations that can exist for patient X's health conditions are identified by medical experts and added to the PHM. These situations can either be specific to patient X or generalized to all patients. For example, if patient X is hypertensive, the situations of interest as defined by physicians and nurses will include *Stage 1*, *Stage 2* and *Danger*. Each of these situations is triggered when a blood pressure reading within a particular range is detected.
4. The adaptations and reactions for the situations identified for patient X's health conditions are identified. For each adaptation or reaction, there must exist one or more actuators to execute them. For example, patient X with dementia health condition can have the "Pick Up Patient" reaction when patient X has wandered off from a range of locations and will require the "Emergency Service" actuator to execute the reaction. Expert knowledge is important in automating this set because it is domain specific and unique for each patient.
5. The personal information of the patient X and the privacy policies patient X wants to enforce on the information that is being stored are also added to patient X's PHM. For example, patient X might decide to share only information about medications with the pharmacist and by so doing, restricting the pharmacist from having access to other information such as medical history. This can either be done by creating roles, assigning users to them and granting access to individual roles or granting access to individual users.
6. After completing the previously mentioned steps, the PHM for patient X is considered as being complete. It is then persisted in the cloud with adequate security measures to prevent the previously mentioned vulnerabilities from affecting the cloud-based system.
7. The previously mentioned sensors and actuators are also linked physically to the architecture. As previously discussed, sensors and actuators require communication channels such as 3G or Wifi to connect to other components. Therefore, it is required the network is set up and that the sensors and actuators have access to it.
8. When a new reading is discovered by one of the sensor devices hooked to the network, the Controller Module (CM) then proceeds to choose the health situation that semantically matches the context derived from the reading by evaluating the context against the table existing for situations. After this is done, the Controller Module can then execute the associated adaptations and reactions through the actuators.

Chapter 5

Detailed Design

In this chapter, the detailed design of the healthcare architecture is presented. The detailed design is interface-driven and this approach allows for flexibility in developing the actual system in whatever language the developer plans to use. In the following sections, we present the implementation details of the modules described in Chapter 4. For each of the modules, we define the properties and operations for the interfaces.

5.1 Data Interfaces

We start with the interfaces of the data structures being processed by each module component. In other words, they provide the implementation details about how parameters used by the modules are represented in the architecture. It is important to address these first because they are used extensively to encapsulate information across the whole system.

5.1.1 ISensor

The ISensor interface represents the data representation for each sensor device. We expect all sensors that are used with the architecture to be modeled and stored using this interface.

- **id**: This is the unique id of the sensor device in the network. When a sensor is registered in the network, it is assigned a unique random id that can be used to identify the sensor later. To ensure that this id is globally unique throughout the architecture, it is implemented as a GUID which represents a 128-bit integer number that is commonly used in enterprise software development especially where an identifier that is globally unique is required.
- **type**: This is the parameter type the sensor is expected to monitor. For example, temperature or

location are different types. We have represented the **type** as a *string*. It is important to keep track of the type the sensor is monitoring in order to have an idea of how to validate its readings. Validating readings from a sensor monitoring *location* will require a different logic from when validating readings from a sensor monitoring *temperature*.

- **unit**: This is the expected unit in which the sensor is expected to report the value of the parameter it is monitoring. An example is Celsius represented as $^{\circ}\text{C}$ for *temperature* parameter type. Keeping track of the **unit** is necessary for validation and data conversion purposes. It is also important to keep track of the unit in order to validate that the sensor sending the reading is the consistent with the one registered in the system. The unit parameter is represented as a *string*.

5.1.2 ISensorReading

The ISensorReading interface is the data representation for each reading from a particular sensor device at any point in time. We expect that all sensors that are registered with network will send readings as represented in this interface.

- **sensor**: This is the details about the sensor sending the reading represented as the **ISensor** interface. It is required to send this with every reading for validation purposes. Without it, it becomes difficult to ensure that the reading originates from a registered sensor which violates **Authenticity** and **Integrity** security requirements for the architecture.
- **value**: This is the tag value of the reading being sent by the sensor. In our architecture, we have implemented it as a string. This is to accommodate for readings that cannot be represented as a number. For example, a location reading involving latitude and longitude readings is usually represented as **+40.7486,-77.8786** where **+40.7486** represents the latitude and **-77.8786** represents the longitude. The system can then use the type of the sensor to understand how to parse and interpret the reading appropriately.
- **time**: This is the timestamp in Coordinated Universal Time (UTC) that the reading was taken by the sensor. It is implemented as a UTC because it is widely accepted and used in architectures worldwide.

5.1.3 IContext

The IContext interface represents the data representation for each context derived from each **SensorReading** data representation after it has been properly validated. In addition, a look up of the patient associated with the record has been done and associated to the reading.

- **patientid**: This is the unique identifier of the patient associated with the sensor device sending the information represented as a GUID. This identifier is derived by doing a look up of the association

between sensors and patients.

- **sensor**: As described in the **ISensorReading** interface, this is the representation of the sensor associated with this reading. The difference at this point is that the sensor has been validated to be registered in the network. In order to do this, a look up of the sensor's identifier matching with the associated type and unit is done. In cases where any discrepancies are discovered, the reading is dropped.
- **value**: As described in the **ISensorReading** interface, this is the **value** of the sensor reading. The difference at this point is that this value has been validated against the expected range of values for the type measured by the sensor. For example, in the case study of a diabetic patient and a sensor reporting blood sugar level in mmol/L, a value of 12 is not acceptable and therefore will be dropped.
- **time**: As described in the **ISensorReading** interface, this is the timestamp the reading was recorded in UTC. The difference at this point is that the timestamp has been validated. In the architecture, any reading that varies more than 16 minutes when compared with the current UTC is dropped.

5.1.4 IProvider

The IProvider interface is the data representation for the health providers for a particular patient in the system. For example, nurses and emergency services are two types of service providers. It is usually stored as part of the health model for the patient.

- **id**: This is the unique identifier of the provider represented in GUID format. This identifier is generated when the provider is added to the system.
- **type**: This is the text description of the type of provider represented as a string. For example, a doctor can be represented as a *doctor* type and a pharmacist can be represented as a *pharmacist* type.
- **name**: This is the text description of the healthcare provider's name represented as a string. For example, the name of the doctor is a string type. In cases where this interface is used to represent a set of providers, the name could be the group name, for example the name of an emergency service provider company.
- **phone**: This is the contact phone number of the healthcare provider, represented as a string. This data is necessary in order to notify the provider about the current health context for the patient.
- **email**: This is the email address of the healthcare provider, represented as a string. This data is equally important in order that the architecture might have different actuator types to convey messages to the provider.

5.1.5 IBiodata

The IBiodata interface is the data representation for the biographical data (biodata) of the patient. It is usually stored as part of the health model.

- **name:** This is the full name of the patient represented as a string. For example, a patient with *Smith* as first name and *Jack* as last name will have his or her full name represented as *Jack Smith*.
- **dateofbirth:** This is the date of birth of the patient represented as a string. For example, the date of birth of a patient born on March 6, 1970 can be represented as *1970-03-06*.
- **address:** This is the full physical address of the patient represented as a string.
- **gender:** This is the gender of the patient represented as a string. It can either be *Male* or *Female*.
- **phone:** This is the phone number of the patient represented as a string. This is necessary in case the patient needs to be contacted through phone.
- **email:** This is the email of the patient represented as a string. This is necessary in case the patient needs to be contacted through email.
- **city:** This is the name of the city where the patient lives, represented as a string.
- **state:** This is the name of the state where the patient lives, represented as a string.
- **country:** This is the name of the country where the patient lives, represented as a string.
- **postalcode:** This is the postal code of the location where the patient lives, represented as a string.
- **bloodtype:** This is the blood type of the patient, represented as a string.
- **language:** This is the primary language that the patient speaks represented as a string. This can be useful while performing internationalization which involves adapting messages and notification to local languages.
- **allergies:** This is the list of all the allergies existing for the patient. Each allergy is represented as a string.
- **notes:** This is the special information that should be noted while providing healthcare services to the patient.

5.1.6 IGuardian

The IGuardian interface is the data representation for the guardians of the patient. It is usually stored as part of the health model for the patient.

- **name**: This is the full name of the guardian of the patient, represented as a string.
- **address**: This is the physical address of the guardian, represented as a string.
- **phone**: This is the phone number of the guardian, represented as a string. This is useful in cases where the guardian needs to be contacted through phone.
- **email**: This is the email of the guardian, represented as a string. This is useful in cases where the guardian needs to be contacted through email.
- **relationship**: This is the relationship between the guardian and the patient, represented as a string.
- **language**: This is the primary spoken language for the guardian represented as a string. This is important for interactions with the guardian.

5.1.7 IPolicy

The IPolicy interface is the data representation for the policies a patient has set. The policies govern how his or her personal information can be shared. This interface is stored as part of the patient's health model and is used by the patient to restrict or grant access to individuals or group of individuals.

- **id**: This is the unique identifier of the individual or group of individuals for which the patient has created access controls. For example, this can be the identifier of a provider or a guardian associated with the patient.
- **accesslist**: This is the data store of all the categories of information the provider or guardian has access to. It can either be an individual record like name or a group of records, for example medical history categorized under the tag name **histories**.

5.1.8 ISituation

The ISituation interface represents the data representation for each situation that can exist for each medical condition based on the current context.

- **patientid**: This is the identifier of the patient associated with the situation.
- **name**: This name is to describe the situation, represented as a string. For example, *Stage 1* or *Stage*

2 are names describing the levels of blood pressure for a hypertensive patient.

- **condition:** This is the medical condition linked to this situation represented as a string. Examples include *diabetes*, *dementia* and *hypertension*.
- **type:** This is the parameter type associated with the situation, represented as a string. An example is *location* type that can trigger a *Danger* situation for a dementia patient.
- **unit:** This is the unit for the parameter type associated with the situation represented as a string. For example, *Celsius* is a unit for *temperature* parameter type.
- **minimumvalue:** This is the minimum value for the parameter type for which this situation can be triggered represented as a string. Since comparison is usually done over a range of values, this value serves as the least value for which this situation can be triggered.
- **maximumvalue:** This is the maximum value for the parameter type for which this situation can be triggered represented as a string. Since comparison is usually done over a range of values, this value serves as the highest value for which this situation can be triggered.

5.1.9 IAdaptation

The IAdaptation interface is the data representation for each adaptation existing for a patient. It is stored as part of the health model for the patient.

- **condition:** This is the medical condition linked to each adaptation represented as a string.
- **situation:** This is the name of situation linked to each adaptation represented as a string.
- **reactions:** This is the key value store of the existing reactions for this adaptation. The key is the name of the reaction to execute. Examples of reactions include “Send Email” or “See Doctor”. The value is the list of the identifiers for all the providers and guardians associated with the reaction.

5.1.10 IHealthModel

The IHealthModel interface represents the data representation for the health model of a particular patient. This model combines other data interfaces associated with the patient as discussed below.

- **id:** This is the unique identifier of the patient throughout the whole system represented as a GUID. This identifier is generated when a new patient is added to the system and remains unchanged throughout the lifecycle of the architecture.
- **biodata:** This is the biodata information of the patient represented as the IBiodata interface.

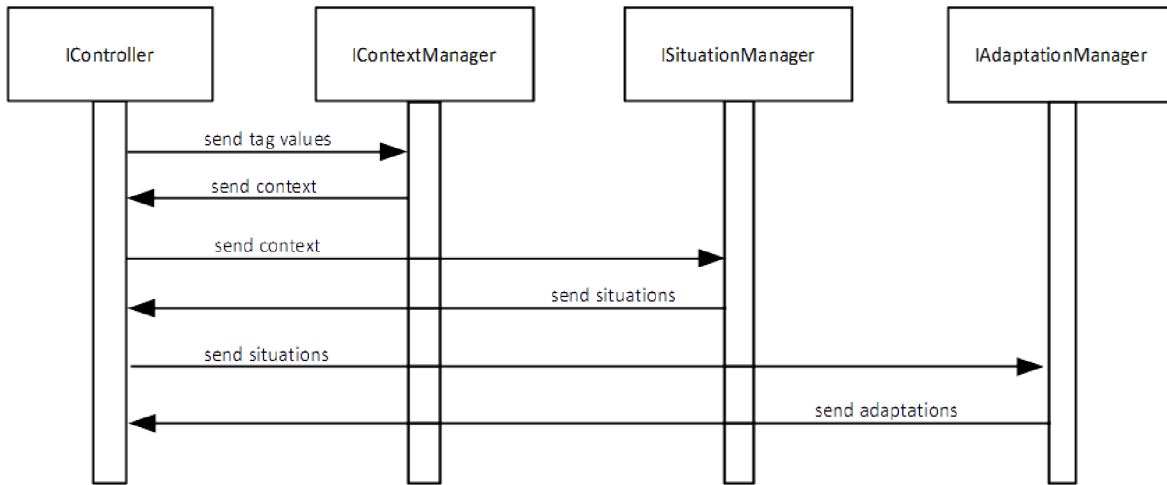


Figure 5.1: Sequence Diagram

- **policies:** This is the list of all the policies existing for the patient with each represented as the `IPolicy` interface.
- **adaptations:** This is the array store of all the adaptations existing for the patient with each adaptation represented as the `IAdaptation` interface.

5.2 Module Interfaces

We continue by providing information about the module interfaces. The architecture contains Controller Module, Context Manager Module, Situation Manager Module and Adaptation Manager Module. Each of these modules is designed to allow maximum flexibility and increase modularity. We discuss the responsibilities and the design of the operations for each component below with the Pseudo code described in Appendix C.

5.2.1 IContextManager

This is the interface for the Context Manager Module as discussed below.

- **compareSensors:** This operation is responsible for comparing the sensor information sent with the reading with the sensor registered with the architecture. In other words, it validates the id, type and unit of the sensor. This is to achieve the *Integrity* and *Authenticity* security requirements of the architecture. It expects `ISensorReading` and `ISensor` interfaces as inputs and returns a *boolean* data type indicating if the comparison was successful.
- **validateSensorReading:** This operation is responsible for validating sensor readings. It validates

that the reading was sent within the last 15 minutes. This validation is helpful in meeting the *Availability* security requirement for the architecture. It subsequently calls **compareSensors** to validate the sensor sending the information. It then checks the parameter type of the reading to invoke the right logic for performing data validation. It expects *ISensorReading* interface as input and returns a type of *IContext*.

5.2.2 ISituationManager

This is the interface for the Situation Manager Module as discussed below.

- **getPatientSituations**: This operation is responsible for retrieving all the situations associated with a patient. It simply does a look up of situations based on the patientid. It expects *IContext* as input and returns a list of *ISituation* as output.
- **getContextSituations**: This operation is responsible for retrieving all the situations associated with a particular context. It invokes the **getPatientSituations** to get all the situations linked to the patient and filters the outcome down based on context parameter type and unit. It expects *IContext* as input and returns a list of *ISituation* as output.

5.2.3 IAdaptationManager

This is the interface for the Adaptation Manager Module as discussed below.

- **fetchSituationAdaptations**: This operation is responsible for fetching the adaptations for a particular situation. It does a look up based on the medical condition and the situation name. It accepts an input list of *ISituation* and returns an output list of *IAdaptation*.
- **hasAccess**: This operation is responsible for checking if a guardian or provider has access to a particular data record. It checks the privacy policies set by the patient in order to ensure that the patient has given explicit access to the guardian or provider for access. This is to achieve the *Confidentiality* security requirement of the architecture. It accepts the *UUID* identifier for the guardian or provider and the *String* description of the access type as inputs and returns *boolean* as output.

5.2.4 IController

The interface for the Controller Module is described below.

- **run**: This operation is called when the Sensor Actuator Network sends sensor readings to the Controller Module. It is responsible for invoking the necessary operations on the information received. It returns a *boolean* to indicate to the Sensor Actuator Network its outcome.

- **createContext**: This operation is responsible for creating IContext out of the sensor readings sent to the Controller Module. In order for this operation to be able to do this, it must invoke the necessary operation on the Context Manager Module. Its return type is IContext.
- **fetchSituations**: This operation is responsible for fetching all the associated situations relating to the IContext derived from the **createContext** operation. It invokes the necessary operations on the Situation Manager Module in order to perform its operations. Its return type is a list of ISituation.
- **fetchAdaptations**: This operation is responsible for fetching all the adaptations that are linked to the situations derived from the **fetchSituations** operation. It invokes the necessary operations on the Adaptation Manager Module in order to perform its operations. Its return type is a list of IAdaptation.
- **executeReactions**: This operation is responsible for running the derived reactions. It returns a *boolean* to indicate the outcome of the operation.

5.3 Helper Interfaces

We continue by providing information about the helper interfaces below.

5.3.1 ILogger

The interface for the Logger Module is described below.

- **log**: This operation is responsible for saving messages regarding the execution of other operations. It expects the priority level of the message and the message as a string inputs and returns *void* as output. This is in accordance to the *Accountability* and *Auditability* security requirements.

Chapter 6

Implementation

In this chapter, the implementation of the context-aware architecture is presented. The case study of a dementia patient is for realizing the architecture is discussed with a thorough explanation of the implemented modules. Finally, the benchmark results of the implemented prototype are shown in the concluding part of this chapter.

6.1 Implementation Platform

The platform used for the implementation is Java using the Java Development Kit(JDK) which includes a virtual machine to compile source code into a working application. We had the option to use the Microsoft .NET, however the .NET is officially supported only on the Windows platform. There have been attempts to port the .NET platform to other platforms for example the free and open source project Mono which intends to make .NET capabilities on the Linux platform[10]. However, these attempts have been hampered with different controversies and performance issues.

Another factor that encouraged us to choose the Java Development Kit is the Integrated Development Environment (IDE) used in the implementation. The .NET platform is usually developed using the Visual Studio[12] platform which requires paid license before being used. However, with the Java Development Kit, we have a variety of development environment such as Netbeans, Eclipse and JBuilder which are both free and efficient.

Due to the previously mentioned advantages that the Java platform provides, we decided to choose it for the implementation of our architecture.

6.1.1 JDK Characteristics

The JDK is the implementation of the Java on mobile, desktop and web developed by Sun Microsystems. It includes a runtime environment, a compiler, an archiver, a documentation generator, an interpreter and other tools needed for implementing Java applications which require database connectivity, Graphic User Interface (GUI) and cryptography. The main motivation behind the JDK is the development of a Software Development Kit(SDK) that can be ported to different platforms with minimal efforts. The first version of the JDK referred to as Oak was released in 1996 and the latest version is Java SE 8.

The portability property of the JDK is achieved through the use of a JVM. A virtual machine is software simulation of a hardware component created for the sole purpose of executing programs. The JVM which is platform dependent is executed by the computer's Central Processing Unit (CPU) in order to process the platform independent Java bytecode generated by the Java compiler. In other words, while writing Java code, the developer doesn't particularly take into consideration the target machine as the JDK does the underlying translation. This is possible because the implementation of all JVMs follow a strict standard that states how Java bytecode can interact with the JVM. Another important responsibility of the JVM is garbage collection which is the automatic management of memory for the application thereby preventing memory related issues unlike languages like C and C++ where the developers are required to manually free up resources.

Security in the JDK is achieved through the use of a sandbox model which allows for the execution of untrusted code in isolation[45]. The JDK achieves this by making it easier for the programmer to write safer code, verifying the Java source code for type safety and restricting access to vital system resources.

6.2 CAHA Implementation

CAHA is implemented using Java on the JDK 8. The components of the architecture are implemented as follows:

- **CAHA.Core:** This module contains the default representation of the *Controller* component of the architecture that acts as the link for routing data and commands among all the other components and the exceptions thrown from this component. The *Controller* component is also responsible for managing the life cycle of other components.
- **CAHA.Data:** This module contains the implementation of the data structures used in the architecture. Each component is developed to meet up with the *JavaBean* specification which indicates how software components can be modeled in such a way that they can be easily persisted and reused later. Each component is serializable and provides access to its properties through *getter* and *setter*

methods. *Getter* methods are used to retrieve private variables while *Setter* methods are used to manipulate private variables.

- **CAHA.Util:** This module contains the implementation of the helper functions used in the architecture and the associated exceptions. Helper classes usually provide methods for other classes to execute common actions. An example is logging responses.
- **CAHA.Models:** This module contains the implementation of the application logic of the architecture and the associated exceptions. This includes the rules for validating and accessing data and logic for executing business rules.

6.3 Case Studies

Two case studies were developed using the CAHA and implemented as described above. The first case study is a patient suffering from dementia that requires constant location monitoring and the second case study is a patient suffering from hypertension that require constant blood pressure monitoring. These two case studies are very useful in understanding the architecture.

6.3.1 Dementia Patient Case Study

Dementia is loss of core brain functions which can range from loss of memory to loss of visual perception. It is usually due to gradual death of brain cells and nerve connections. Dementia currently affects 26 million people globally[80]. Furthermore, according to [6], about 6-15% of elderly patients in Canada suffer a form of dementia and the number is expected to double to 2031. Currently, dementia has no cure[80].

In this case study, we intend to provide constant monitoring of longitude and latitude location values of dementia patients. We hope that caregivers and ambulance services will be able to use the system to prevent wandering by dementia patients.

6.3.1.1 System Design

In order to implement the system, we need to define the components we need.

- **Sensors:** In order to constantly monitor location values, we identified that we need a location sensor that is capable of measuring and reporting location in latitude and longitude formats. For our purposes, we identified that a GPS sensor will be most appropriate for this task.
- **Actuators:** In this case, we have two actuators, the emergency services and the caregiver assigned to the patient.

- **Situations:** We used **Concordia University Loyola Campus** as a sample acceptable location. The GPS location value from Google Maps in latitude longitude format is **45.458580, -73.640283**. The situations defined here reflect *Warning* and *Danger* situations. The *Warning* situation refers to the state where the location varies by a value between 0.0009 and 0.01. We present it in the Situation Expression Language as follows.

$$Warning : \{(0.0009 < \$AbsDiff[Location] < 0.01)\} \quad (6.1)$$

where “*AbsDiff*” is a situation extender that provides the ability to extend the reasoning capability of the framework by triggering custom user defined functions that takes context information as an input and returns an absolute value. “*AbsDiff*” takes in the current location and compares it with the base location.

The *Danger* situation refers to the state where the location varies by a value between 0.01 and above which we defined as the *Danger* situation. For the *Danger* situation, we represent it in the Situation Expression Language as follows.

$$Danger : \{(\$AbsDiff[Location] >= 0.01)\} \quad (6.2)$$

- **Adaptations:** We define the following appropriate adaptations to the two situations defined above.
 1. The adaptation to the *Warning* situation is *Notify Caregiver*. This adaptation sends a notification to the caregiver to pick up the patient from the exact current location since the location is closer to the base location. The workflow of this adaptation is presented using the Workflow Expression Language is presented below.

$$Exec(Notify["Caregiver", PatientId, Location]); \quad (6.3)$$

2. The adaptation to the *Danger* situation is *Notify Emergency*. This adaptation sends notification to the emergency services to pick up the patient from the exact current location. The workflow of this adaptation is presented using the Workflow Expression Language is presented below.

$$Exec(Notify["Emergency", PatientId, Location]); \quad (6.4)$$

6.3.1.2 System Process Model

The system process model contains the sequence of the following steps:

1. **System Activation:** The system is activated when the sensor registers a new location reading which happens at certain intervals.

2. **Sensor Reading Validation:** After the sensor sends the new reading to the *Controller* module, the *Controller* module sends it to the *ContextManager* for validation. The validation steps are as follows.

- checking that the source sensor is registered with the system
- checking that the reading was sent within the last 5 minutes in order to prevent replay attacks
- checking that the reading is within the acceptable range of values. In this case study, latitude values can only range from -90 to +90 and longitude values can only range from -180 to +180. Therefore, we can safely filter out any values outside these ranges.

The process ends by returning the validated sensor reading that we refer to as **context** to the *Controller* module.

3. **Situations Association:** After the context is passed back to the *Controller* module, the *Controller* module requests for the associated situations related to the context.

4. **Adaptation Resolution:** After the related situation(s) have been identified, the *Controller* module then contacts the *AdaptationManager* to resolve the adaptations existing for the identified situation(s) taking into consideration the privacy policies for that patient. In this case study, the reactions are simply to either notify the care giver or the emergency services. The *Controller* also makes use of the *Logger* to keep track of the context, situations realized and the reactions executed for the patient.

6.3.2 Hypertension Patient Case Study

Hypertension is a medical condition where an individual suffers from an abnormally high blood pressure. Blood pressure is usually measured in two dimensions: the systolic pressure (the pressure at which blood is pumped around the body) and the diastolic pressure (the pressure at which the heart is being refilled with blood). Under normal circumstances, blood pressure values are within the ranges of 100 to 140mmHg systolic and 60 to 90mmHg diastolic. When a value higher than these ranges is recorded, the condition is referred to as Hypertension.

In this case study, we intend to provide constant monitoring of blood pressure for hypertensive patients. We hope that caregivers, doctors and emergency services will be able to use the system to assist the treatment of hypertensive patients.

6.3.2.1 System Design

In order to implement the system, we need to define the components we need.

- **Sensors:** In order to constantly measure blood pressure values, we need a sphygmomanometer that

can be used to either manually or automatically report the blood pressure values of a patient in mmHg format.

- **Actuators:** In this case, we have four actuators, the patient, the caregiver, the doctor and the emergency services.
- **Situations:** There are three main situations that can be realized from the blood pressure readings explained as follows:

1. **Stage 1:** This is the case when a systolic value between 140 and 159mmHg or a diastolic value between 90 and 99mmHg is recorded. We present it in the Situation Expression Language as follows.

$$Stage\ 1 : \{(140 < BP_{systolic} < 159) OR (90 < BP_{diastolic} < 99)\} \quad (6.5)$$

2. **Stage 2:** This is the case when a systolic value between 160 and 179mmHg or a diastolic value between 100 and 109mmHg is recorded. We present it in the Situation Expression Language as follows.

$$Stage\ 2 : \{(160 < BP_{systolic} < 179) OR (100 < BP_{diastolic} < 109)\} \quad (6.6)$$

3. **Danger:** This is the case when a systolic value between 180 and above or a diastolic value between 110 and above is recorded. We present it in the Situation Expression Language as follows.

$$Danger : \{(180 < BP_{systolic}) OR (110 < BP_{diastolic})\} \quad (6.7)$$

- **Adaptations:** We define the following appropriate adaptations to the three situations defined above.

1. The adaptations to the *Stage 1* situation is *Notify Caregiver* and *Notify Patient*. *Notify Caregiver* sends a notification to the caregiver indicating that the patient is currently experiencing *Stage 1* situation. The workflow of this adaptation is presented using the Workflow Expression Language is presented below.

$$Exec(Notify["Caregiver", PatientId, "Stage1"]); \quad (6.8)$$

The second adaptation involves sending a notification to the patient. The notification usually is a message advising the patient on the appropriate action to take for example "Take medication". The workflow of this adaptation is presented using the Workflow Expression Language is presented below.

$$Exec(Notify["Patient", PatientId, "Stage1"]); \quad (6.9)$$

2. The adaptation to the *Stage 2* situation is *Notify Doctor*. This adaptation sends notification to the doctor that the patient is currently experiencing *Stage 2* situation. The workflow of this adaptation is presented using the Workflow Expression Language is presented below.

$$Exec(Notify["Doctor", PatientId, "Stage2"]); \quad (6.10)$$

3. The adaptation to the *Danger* situation is *Notify Emergency*. This adaptation sends notification to the emergency services that the patient is currently experiencing *Danger* situation. The workflow of this adaptation is presented using the Workflow Expression Language is presented below.

$$Exec(Notify["Emergency", PatientId, "Danger"]); \quad (6.11)$$

6.3.2.2 System Process Model

The system process model contains the sequence of the following steps:

1. **System Activation:** The system is activated when the sphygmomanometer registers a new blood pressure value.
2. **Sensor Reading Validation:** The validations done in this stage includes
 - checking that the source sensor is registered with the system
 - checking that the reading was sent within the last 5 minutes in order to prevent replay attacks
 - checking that the reading is within the acceptable range of values. In this case study, any reading reporting a systolic value over 500mmHg or a diastolic value over 500mmHg is discarded.
3. **Situations Association:** After the context is passed back to the *Controller* module, the *Controller* module requests for the associated situations related to the context.
4. **Adaptation Resolution:** After the related situation(s) have been identified, the *Controller* module then contacts the *AdaptationManager* to resolve the adaptations existing for the identified situation(s) taking into consideration the privacy policies for that patient. The *Controller* also makes use of the *Logger* to keep track of the context, situations realized and the reactions executed for the patient.

6.4 Tests And Results

In order to ascertain the validity of our implementation, we wrote unit test cases for each component from a white box perspective. Unit testing helps in ascertaining that the logic was properly implemented and that

there are no side effects to the code written. The normal approach to unit testing is to create drivers that serve as calling unit and to stub dependencies for each component. By doing this, code can easily be isolated and investigated for errors. The test cases we chose for each component were suitable enough to catch all the success and failure scenarios.

After writing the unit tests, we wrote the feature tests to verify that the system is able to perform its functions as a whole unit. Through these feature tests, we were able to identify and rectify design issues. We also used static code analysis tool in Netbeans IDE to analyze source code in order to improve the code quality and readability.

We also performed security audit testing to ascertain that the privacy policies were enforced. We were able to achieve this through the use of specialized tools that can test for vulnerabilities in enterprise systems. The security features tested included:

- ensuring that each actuator only has access to the exact amount of information as defined by the patient’s privacy policies. In order to achieve this, we simulated that the patient in the hypertension case study has placed a restriction on emergency services being able to access his medications.
- ensuring that the architecture is not susceptible to denial of service attacks thereby satisfying the *Availability* requirement. We were able to test this on the *Controller* module using Apache JMeter.
- ensuring that there is a log of each activity performed by the system thereby satisfying the *Accountability* and *Auditability* requirements. We verified this by manually examining the logs after running the case studies.
- ensuring that the information processed by the architecture is accurate and communicated in a secure manner thereby satisfying the *Confidentiality* and *Integrity* requirements. We tested this by using Wireshark to examine the traffic between the Sensor Actuator Network and the Cloud Cyberspace. We also examined the communication between the Cloud Cyberspace and the Service Providers. By doing this, we ascertained that the information flow is encrypted and therefore can’t be intercepted by malicious attackers.
- ensuring that only registered sensors can send data thereby satisfying the *Authenticity* requirement. We simulated sending request from a sensor that has not been registered previously in the system and noticed that the request was dropped.

Chapter 7

Conclusion And Future Work

This thesis aims at providing a context-aware architecture that can be used for developing healthcare applications especially for elderly patients. This architecture can be used by developers of healthcare systems to provide real time services to patients. In developing such a system, developers need to (1) identify sensors and actuators related to the targeted medical conditions, (2) define the different contexts that can be measured by these sensors, (3) define the situations of interest for each targeted medical condition based on the context and (4) define the policies and adaptations. In the following summary we emphasize the major contributions in the thesis as well as its limitations, provide some suggestions for future work, and comment on the challenges to overcome in further extensions.

A literature review in Chapter 2 presents the current healthcare systems and their inadequacies in providing quality healthcare services to patients. Overcoming these inadequacies is a primary motivation to propose a new architecture in which security and privacy policies are integrated to enhance the quality of services. Given that healthcare systems are being largely deployed worldwide, there is definitely a need for an architecture that can be used as the skeleton for such systems. Our research has produced one such architecture.

An overview of the concepts used by context-aware healthcare architectures is provided in Chapter 3. Chapter 4 gives a generalized overview of our proposed component based architecture. In Chapter 5, we explained the data and module interfaces for the architecture. An implementation of the architecture is illustrated with case studies of dementia and hypertensive patients in Chapter 6. To ascertain the validity of the architecture, we have provided information about our tests and benchmarks. Some of the virtues of our architecture are described below.

- We used CBSE principles in constructing the architecture. It is an assemblage of components, where each component provides/receives services through its well-defined interfaces. Two components are

connected only if services provided by one can be received by the other, subject to security and privacy constraints.

- Each component can be tested both individually as well as in conjunction with the components with which it interacts. The functionality at its interfaces, and the conformance of policies for invoking the functionality can be tested.
- By enriching an interface specification and by adding new interfaces the architecture is allowed to grow. That is, new components can be added to the current architecture in order to accommodate new requirements. As an example, “health insurance entity”, if required, can be added to the architecture without violating the existing architectural connections. Thus, the system grows “conservatively” meaning the overall behavior of the extended system absorbs (without violation) the behavior of the original architecture.
- Situation and Workflow languages are also extendable, in the sense new operations can be added within their current grammatical structures. We can add macros, pre-compiled predefined functions, to speed up the triggering of reactions.
- We have defined actuator types in order that any number of actuators within each type can be attached to the system. At run time the actuator controller will choose one actuator that is available to execute reactions. Moreover, it will choose actuators necessary to execute multiple actions as specified in the WPL language. This is possible because we can define a mapping from WPL expressions to actuator types.

7.1 Limitations

There are a few limitations in our implementation and they are listed below.

- We simulated the *Elderly Home* and *Service Provider* components of the entity model in our implementation due to resource limitations and restriction involved in setting up real life sensor networks.
- We considered basic examples of health context, situations and adaptations.
- We have not done performance testing of how the implementation would behave in real life scenarios.
- We only implemented basic security and privacy mechanisms because some of the requirements are out of the scope of this work. An example is non-repudiation.
- We considered a small array of service providers. Additional service providers that are of interest to the architecture include Psychologists, Health Insurance Agencies, and Government Agencies.

7.2 Future Work

The future work that can be done for this architecture are described below.

- The entity model can be expanded to include hardware-based or software-based actuators which are different from the human-based actuators considered in our implementation.
- Complex examples of health context, situations and adaptations can be considered. A good place to start is to consider patients suffering from multiple health conditions. In order to achieve this, there is a need for expert knowledge on adaptations for complex health situations.
- The security requirement of non-repudiation can be investigated further and then implemented in the architecture in a more complex fashion.
- The current architecture and its implementation can be scaled up to accommodate new resources and knowledge and can ported to other platforms.
- The architecture can be implemented and tested with real life sensors. In order to achieve this, a collaboration with medical professionals is necessary.

Chapter 8

Bibliography

- [1] Care coordination and communication software for senior care — caremerge. <http://www.caremerge.com/>. [Online; accessed 29/01/2015].
- [2] Care predict — empowering independent living for seniors. <http://carepredict.com/>. [Online; accessed 29/01/2015].
- [3] Chinese aged diabetic assistant. <http://healthmarketinnovations.org/program/chinese-aged-diabetic-assistant-cada>. [Online; accessed 02-November-2015].
- [4] Cloud computing: What is infrastructure as a service. <https://technet.microsoft.com/en-us/magazine/hh509051.aspx>. [Online; accessed 29/01/2015].
- [5] Corba. <http://www.omg.org/corba/>. [Online; accessed 29/01/2015].
- [6] Dementia. <http://healthycanadians.gc.ca/diseases-conditions-maladies-affections/disease-maladie/dementia-demence/index-eng.php>. [Online; accessed 02-November-2015].
- [7] Episurveyor. <http://home.magpi.com/>. [Online; accessed 02-November-2015].
- [8] Getmyrx delivered free today. <https://www.getmyrx.com/>. [Online; accessed 29/01/2015].
- [9] Mobile hiv/aids support. <http://healthmarketinnovations.org/program/mobile-hiv-aids-support>. [Online; accessed 02-November-2015].
- [10] Mono (software) — wikipedia the free encyclopedia. [https://en.wikipedia.org/wiki/Mono_\(software\)](https://en.wikipedia.org/wiki/Mono_(software)). [Online; accessed 02-November-2015].

- [11] Nutrition for older persons. <http://www.who.int/nutrition/topics/ageing/en/index1.html>. [Online; accessed 29/01/2015].
- [12] Visual studio. <https://www.visualstudio.com/>. [Online; accessed 02-November-2015].
- [13] Guidelines for improving the care of the older person with diabetes mellitus, 2003.
- [14] mhealth for development: The opportunity of mobile technology for healthcare in the developing world. <http://www.vitalwaveconsulting.com/pdf/2011/mHealth.pdf>, February 2009. [Online; accessed 29/01/2015].
- [15] mhealth: New horizons for health through mobile technologies. http://www.who.int/goe/publications/goe_mhealth_web.pdf, June 2011. [Online; accessed 29/01/2015].
- [16] Seniors vulnerability report. http://www.theprovince.com/pdf/uw_2011_seniors_vulnerability_report_low-rez__final.pdf, 2011. [Online; accessed 29/01/2015].
- [17] Mladen A Vouk. Cloud computing—issues, research and implementations. *CIT. Journal of Computing and Information Technology*, 16(4):235–246, 2008.
- [18] Vangalur S Alagar, Mubarak Mohammad, Kaiyu Wan, and Sofian Alsalman Hnaide. A framework for developing context-aware systems. *EAI Endorsed Trans. Context-aware Syst. & Appl.*, 1:e2, 2014.
- [19] JZ. Ayanian, JS. Weissman, EC. Schneider, JA. Ginsburg JA, and AM. Zaslavsky. Unmet health needs of uninsured adults in the united states. *JAMA*, 284(16):2061–2069, 2000. [Online; accessed 29/01/2015].
- [20] F. Bari and F. Mark. Security and privacy in mobile health. *CIO Journal*, August 2013.
- [21] L. Benton. Dissecting the mhealth storage dilemma: Cloud or on-site? <http://www.mhealthnews.com/blog/dissecting-mhealth-storage-dilemma-cloud-or-site>, October 2014. [Online; accessed 29/01/2015].
- [22] Bitglass. The 2014 bitglass healthcare breach report, 2014.
- [23] Jesse Michael Blum and EH Magill. The design and evaluation of personalised ambient mental health monitors. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1–5. IEEE, 2010.
- [24] J. Efrim Boritz. {IS} practitioners’ views on core concepts of information integrity. *International Journal of Accounting Information Systems*, 6(4):260 – 279, 2005.

- [25] Everybodys Business. Strengthening health systems to improve health outcomes: Whos framework for action. *Ginebra: OMS*, 2007.
- [26] Business Dictionary.
- [27] A. Carrie. The older population: 2010. Technical Report C2010BR-09, U.S. Census Bureau, November 2011. [Online; accessed 29/01/2015].
- [28] Rajiv Chakravorty. A programmable service architecture for mobile medical care. In —, pages 532–536. IEEE, 2006.
- [29] Giovanni Chiarini, Pradeep Ray, Shahriar Akter, Cristina Masella, and Aura Ganz. mhealth technologies for chronic diseases and elders: a systematic review. *Selected Areas in Communications, IEEE Journal on*, 31(9):6–18, 2013.
- [30] A. Chin, M.J. Paw, J.M. Dekker, E.J. Feskens, E.G. Schouten, and D. Kromhout. How to select a frail elderly population? a comparison of three working definitions. Technical Report 11, Division of Human Nutrition and Epidemiology, Wageningen Agricultural University, The Netherlands, 1999.
- [31] CHSRF. Health systems planning for the aging population: A backgrounder. pages 1–6, 2009.
- [32] CIHI. Health care in canada, 2011: Focus on seniors and aging. pages 1–135, 2011.
- [33] D. Claudette. *Aboriginal elder abuse in Canada*. Aboriginal Healing Foundation Ottawa, Canada, 2002.
- [34] College Of Physicians And Surgeons Of Ontario. Confidentiality of personal health information, April 2006.
- [35] Dorothy W Curtis, Esteban J Pino, Jacob M Bailey, Eugene I Shih, Jason Waterman, Staal A Vinterbo, Thomas O Stair, John V Guttag, Robert A Greenes, and Lucila Ohno-Machado. Smartan integrated wireless system for monitoring unattended patients. *Journal of the American Medical Informatics Association*, 15(1):44–53, 2008.
- [36] B. David and T. Alistair. Community care of vulnerable older people: cause for concern. *British Journal of General Practice*, 63(615):549–550, 2013.
- [37] M. David. The changing elderly population and future health care needs. *Journal of urban health*, 76(1):24–38, 1999.
- [38] Tarshis Debbie. Overview of the personal health information protection act. *Law And Governance, LegalFocus on Healthcare and Insurance*, 8(4), June 2004.

- [39] Bates DW, Leape LL, Cullen DJ, and et al. Effect of computerized physician order entry and a team intervention on prevention of serious medication errors. *JAMA*, 280(15):1311–1316, 1998.
- [40] P. Elizabeth. National survey on abuse of the elderly in canada. *Journal of Elder Abuse & Neglect*, 4(1-2):5–58, 1993.
- [41] S. Espinoza and J. Walston. Frailty in older adults: Insights and interventions. *CLEVELAND CLINIC JOURNAL OF MEDICINE*, 72(12), December 2005. [Online; accessed 29/01/2015].
- [42] Robert G Evans, Morris L Barer, and Theodore R Marmor. *Why are some people healthy and others not?: The determinants of health of populations*. Cambridge Univ Press, 1994.
- [43] Centers for Disease Control (CDC) et al. Healthy aging improving and extending quality of life among older americans. *Chronic Disease Prevention and Health Promotion*, 2009.
- [44] PK. Goel, SK. Garg, JV. Singh, M. Bhatnagar, H. Chopra, SK. Bajpai, et al. Unmet needs of the elderly in a rural population of meerut. *Indian J Comm Med*, 28:165Á6, 2003.
- [45] Li Gong, Marianne Mueller, Hemma Prafullchandra, and Roland Schemers. Going beyond the sandbox: An overview of the new security architecture in the java development kit 1.2. In *USENIX Symposium on Internet Technologies and Systems*, pages 103–112, 1997.
- [46] A.H. Greene. When hipaa applies to mobile applications, June 2011.
- [47] V. Group. Evaluating mhealth adoption barriers: Privacy and regulation. <http://mhealthregulatorycoalition.org/wp-content/uploads/2013/01/VodafoneGlobalEnterprise-mHealth-Insights-Guide-Evaluating-mHealth-Adoption-Privacy-and-Regulation.pdf>, 2013. [Online; accessed 29/01/2015].
- [48] Capital Health. Determinants of health. <http://www.cdha.nshealth.ca/halifax-community-health-board/population-health/determinants-health>.
- [49] HealthCareBusinessTech. Mobile health apps create privacy risk, study says. <http://www.healthcarebusinesstech.com/mobile-health-apps-privacy/>, 2014. [Online; accessed 29/01/2015].
- [50] Cecil G Helman. *Culture, health and illness*. CRC Press, 2007.
- [51] Jong-yi Hong, Eui-ho Suh, and Sung-Jin Kim. Context-aware systems: A literature review and classification. *Expert Systems with Applications*, 36(4):8509–8522, 2009.
- [52] P. Hope, SM. Bamford, S. Beales, K. Brett, D. Kneale, M. Macdonnell, and A. McKeon. Creating

sustainable health and care systems in ageing societies. Technical report, Ageing Societies Working Group, 2012. [Online; accessed 29/01/2015].

- [53] <http://www.helptheaged.ca/>. The importance of health care services for the elderly, 9 2012.
- [54] R.S.H. Istepanian, E. Jovanov, and Y.T. Zhang. Guest editorial introduction to the special section on m-health: Beyond seamless mobility and global wireless health-care connectivity. *Trans. Info. Tech. Biomed.*, 8(4):405–414, December 2004. [Online; accessed 29/01/2015].
- [55] Leonardo Horn Iwaya. *A security framework for mobile health data collection*. PhD thesis, Universidade de São Paulo, 2014.
- [56] Georgios Kambourakis, Eleni Klaoudatou, and Stefanos Gritzalis. Securing medical sensor environments: the codeblue framework case. In *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*, pages 637–643. IEEE, 2007.
- [57] H. Kharrazi, R. Chisholm, D. VanNasdale, and B. Thompson. Mobile personal health records: An evaluation of features and functionality. *International Journal of Medical Informatics*, 81(9):579 – 593, 2012. [Online; accessed 29/01/2015].
- [58] E. Kim and L. Geistfeld. What makes older adults vulnerable to exploitation or abuse?, 2008.
- [59] Jesper Kjeldskov and Mikael B Skov. Supporting work activities in healthcare by mobile electronic patient records. In *Computer Human Interaction*, pages 191–200. Springer, 2004.
- [60] Predrag Klasnja and Wanda Pratt. Healthcare in the pocket: mapping the space of mobile-phone health interventions. *Journal of biomedical informatics*, 45(1):184–198, 2012.
- [61] JeongGil Ko, Jong Hyun Lim, Yin Chen, Rvǎzvan Musvaloiu-E, Andreas Terzis, Gerald M Masson, Tia Gao, Walt Destler, Leo Selavo, and Richard P Dutton. Medisn: Medical emergency detection in sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 10(1):11, 2010.
- [62] Pardeep Kumar and Hoon-Jae Lee. Security issues in healthcare applications using wireless medical sensor networks: A survey. *Sensors*, 12(1):55–91, 2011.
- [63] Ada Kwan. mhealth solutions for improving mental health and illnesses in the aging process, 2013.
- [64] F. Lauren, B. Deana, L. Chien-Chih, B. Samuel, and B. Joseph. Frail elderly as disaster victims: Emergency management strategies. *Prehospital and Disaster Medicine*, 17:67–74, 6 2002.
- [65] Geng Lin, David Fu, Jinzy Zhu, and Glenn Dasmalchi. Cloud computing: It as a service. *IT professional*, (2):10–13, 2009.

- [66] Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger, and Dawn Leaf. Nist cloud computing reference architecture: Recommendations of the national institute of standards and technology (special publication 500-292). 2012.
- [67] Peter Lonsdale, Chris Baber, Mike Sharples, and Theodoros N Arvanitis. A context awareness architecture for facilitating mobile learning. *Learning with mobile devices: Research and development*, pages 79–85, 2004.
- [68] David Malan, Thaddeus Fulford-Jones, Matt Welsh, and Steve Moulton. Codeblue: An ad hoc sensor network infrastructure for emergency medical care. In *International workshop on wearable and implantable body sensor networks*, volume 5, 2004.
- [69] Christian Malo, Xavier Neveu, Michel Patrick Archambault, Marcel Émond, and Marie-Pierre Gagnon. Exploring nurses’ intention to use a computerized platform in the resuscitation unit: Development and validation of a questionnaire based on the theory of planned behavior. *Interact J Med Res*, 1(2):e5, Sep 2012.
- [70] P. Maria, T. Katerina, B. Evangelos, and T. Taxiarchis. Mobile phone application to support the elderly. *International Journal of Cyber Society and Education*, 6(1):51–56, 2013.
- [71] M. McCarthy. Experts warn on data security in health and fitness apps. *BMJ*, 347, 2013.
- [72] L. McCoy and L. Brown. Health status among low-income elderly persons: Rural-urban differences. *Soc. Sec. Bull.*, 41:14, 1978.
- [73] Susan A McDaniel. The conundrum of demographic aging and policy challenges: a comparative case study of canada, japan and korea. *Canadian Studies in Population*, 36(1-2):37–62, 2009.
- [74] A. Melissa, M. Arnold, and R. Kenneth. Social vulnerability, frailty and mortality in elderly people. *PLoS One*, 3(5):e2232, 2008.
- [75] Mubarak Mohammad and Vangalur Alagar. A formal approach for the specification and verification of trustworthy component-based systems. *Journal of Systems and Software*, 84(1):77–104, 2011.
- [76] Fabiane Nardon. Virtual health project. <http://healthmarketinnovations.org/program/virtual-health-pet>. [Online; accessed 02-November-2015].
- [77] HS Ng, ML Sim, and CM Tan. Security issues of wireless sensor networks in healthcare applications. *BT Technology Journal*, 24(2):138–144, 2006.
- [78] Jason WP Ng, Benny PL Lo, Oliver Wells, Morris Sloman, Nick Peters, Ara Darzi, Chris Toumazou,

- and Guang-Zhong Yang. Ubiquitous monitoring environment for wearable and implantable sensors (ubimon). In *International Conference on Ubiquitous Computing (UbiComp)*. Citeseer, 2004.
- [79] Anthony C Norris, RS Stockdale, and S Sharma. A strategic approach to m-health. *Health informatics journal*, 15(3):244–253, 2009.
- [80] World Health Organization et al. Dementia fact sheet. *World Health Organization, Geneva*, 2012.
- [81] Sameer Pai, Sergio Bermudez, Stephen B Wicker, Marci Meingast, Tanya Roosta, Shankar Sastry, and Deirdre K Mulligan. Transactional confidentiality in sensor networks. *Security & Privacy, IEEE*, 6(4):28–35, 2008.
- [82] Shwetak N Patel, Julie A Kientz, Gillian R Hayes, Sooraj Bhat, and Gregory D Abowd. Farther than you may think: An empirical investigation of the proximity of users to their mobile phones. In *UbiComp 2006: Ubiquitous Computing*, pages 123–140. Springer, 2006.
- [83] JD. Payne. The state of standards and interoperability for mhealth among low- and middle-income countries. *mHealth Alliance*, March 2013.
- [84] G. Peat, R. McCarney, and P. Croft. Knee pain and osteoarthritis in older adults: a review of community burden and current use of primary health care. *Annals of the rheumatic diseases*, 60(2):91–97, 2001.
- [85] Salil Pradhan. Semantic location. *Personal Technologies*, 4(4):213–216, 2000.
- [86] PricewaterhouseCoopers. Moneytree report. Technical report, National Venture Capital Association, 2014.
- [87] Province of Alberta. Alberta health information protection act, 2010.
- [88] Christine Zhenwei Qiang, Masatake Yamamichi, Vicky Hausman, and Daniel Altman. Mobile applications for the health sector, December 2011.
- [89] M. Ramon, D. Jaime, and P. Xavier. Security specification and implementation for mobile e-health services. In *e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004 IEEE International Conference on*, pages 241–248. IEEE, 2004.
- [90] Anand Ranganathan and Roy H Campbell. An infrastructure for context-awareness based on first order logic. *Personal and Ubiquitous Computing*, 7(6):353–364, 2003.
- [91] E. Renehan, B. Dow, X. Lin, I. Blackberry, I. Haapala, E. Gaffy, E. Cyarto, and K. Brashe. Healthy ageing literature review 2012. Technical report, State of Victoria, Department of Health, 2012. [Online;

accessed 29/01/2015].

- [92] Vincent Ricquebourg, David Durand, David Menga, B Marine, Laurent Delahoche, Christophe Loge, and Anne-Marie Jolly-Desodt. Context inferring in the smart home: An swrl approach. In *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on*, volume 2, pages 290–295. IEEE, 2007.
- [93] B. Robert and K. Daniel. Response of the elderly to disaster: An age-stratified analysis. *The International Journal of Aging and Human Development*, 16(4):283–296, 1982.
- [94] B. Ronet, D. Heather, and L. Mark. Violence against the elderly a comparative analysis of robbery and assault across age and gender groups. *Research on aging*, 20(2):183–198, 1998.
- [95] Norman Ryder. Notes on stationary populations, 1975.
- [96] SecureWorks. Healthcare security breaches:secure patient information and avoid financial penalties. <http://www.secureworks.com/assets/pdf-store/other/infographic.healthcare.pdf>. [Online; accessed 29/01/2015].
- [97] CD. Speer and SG. Myra. Mental health needs of older adults and primary care: Opportunity for interdisciplinary geriatric team practice. *Clinical Psychology: Science and Practice*, 10(1):85–101, 2003.
- [98] Symantec. Security and privacy for healthcare providers. *Best Practices Series For Healthcare*, 2009. [Online; accessed 29/01/2015].
- [99] Symantec. Internet security threat report 2014. Technical report, Symantec, 2014. [Online; accessed 29/01/2015].
- [100] Center For Technology and Aging. Technologies to help older adults maintain independence:advancing technology adoption, July 2009.
- [101] The Economist Intelligence Unit Limited. Healthcare strategies for an ageing society. Technical report, Economist Intelligence Unit, 2009. [Online; accessed 29/01/2015].
- [102] The Government Of Saskatchewan. Health information protection act, 2003.
- [103] The Markle Foundation. Markle common framework, 2006.
- [104] K. Thomas and D. Thomas. Assessing long-term impacts of a natural disaster: A focus on the elderly. *The Gerontologist*, 19(6):555–566, 1979.

- [105] M. Tueth. Exposing financial exploitation of impaired elderly persons. *The American Journal of Geriatric Psychiatry*, 8(2):104 – 111, 2000. [Online; accessed 29/01/2015].
- [106] U.S. Department of Health and Human Services. The nationwide privacy and security framework for electronic exchange of individually identifiable health information, December 2008.
- [107] US Department of Health and Human Services and others. Older adults and mental health: Issues and opportunities. *Rockville, MD: Author*, 2001.
- [108] Kaiyu Wan, Vasu Alagar, and Joey Paquet. An architecture for developing context-aware systems. In *Modeling and Retrieval of Context*, pages 48–61. Springer, 2006.
- [109] Lizhe Wang, Gregor Von Laszewski, Andrew Younge, Xi He, Marcel Kunze, Jie Tao, and Cheng Fu. Cloud computing: a perspective study. *New Generation Computing*, 28(2):137–146, 2010.
- [110] D. Whalen. The mental health needs of an aging population. <http://www.cmhanl.ca/pdf/Healthy%20Aging.pdf>. [Online; accessed 29/01/2015].
- [111] WIKIA. Markle common framework for private and secure health information exchange, 2015.
- [112] Anthony Wood, Gilles Virone, Thao Doan, Qihua Cao, Leo Selavo, Yafeng Wu, L Fang, Zhimin He, Shan Lin, and Jack Stankovic. Alarm-net: Wireless sensor networks for assisted-living and residential monitoring. *University of Virginia Computer Science Department Technical Report*, 2, 2006.
- [113] Anthony D Wood, John Stankovic, Gilles Virone, Leo Selavo, Zhimin He, Qihua Cao, Thao Doan, Yafeng Wu, Lei Fang, Radu Stoleru, et al. Context-aware wireless sensor networks for assisted living and residential monitoring. *Network, IEEE*, 22(4):26–33, 2008.
- [114] World Health Organization. Definition of an older or elderly person. <http://www.who.int/healthinfo/survey/ageingdefnolder/en/>, 2015. [Online; accessed 29/01/2015].
- [115] R. Zhang and L. Liu. Security models and requirements for healthcare application clouds. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 268–275, July 2010.

Appendix A

Situation Expression Language

The *Situation Expression Language* defines the syntactic structure of situation expressions. It allows defining situations based on logical and mathematical operations over dimensions, tag values and other situations. The language supports the following operations.

- Logical *AND*, *OR*, & *NOT* operations between situations.
- *Equal*, *Not Equal*, *Bigger*, *Smaller*, *Bigger or equal*, & *Smaller or equal* between dimensions.
- *Logical AND*, *OR*, & *NOT* operation between dimension expressions.
- The basic arithmetic operations, namely *Addition*, *Subtraction*, *Division*, and *Multiplication* defined on numeric tag values.

The situation expression language is defined using a context free grammar (CFG) as follows.

The root grammar in this CFG is called *Situation*. *Situation* should be defined between curly braces $\{\}$. *Situation* can be either a *Hybrid Situation* which contains a situation expression over dimensions and other situations, or a *Literal Situation* which contains a situation expression over dimensions only. The root grammar is presented as follows.

$$\langle \mathbf{Situation} \rangle ::= \{ \langle \mathbf{SituationRule} \rangle \} | \langle \mathbf{LiteralExpression} \rangle;$$

$$\begin{aligned}
\langle \textit{SituationRule} \rangle ::= & \langle \textit{ANDSituationRule} \rangle | \\
& \langle \textit{ORSituationRule} \rangle | \\
& \langle \textit{NOTSituationRule} \rangle | \\
& \langle \textit{LiteralExpression} \rangle | \\
& \langle \textit{SituationToken} \rangle
\end{aligned}$$

The logical operations used in the situation expression language are AND, OR, & NOT. The operations are defined recursively over the *Situation Rule* notation as follows.

$$\begin{aligned}
\langle \textit{ANDSituationRule} \rangle ::= & \langle \textit{SituationRule} \rangle \textit{ AND } \langle \textit{SituationRule} \rangle \\
\langle \textit{ORSituationRule} \rangle ::= & \langle \textit{Situation} \rangle \textit{ OR } \langle \textit{SituationRule} \rangle \\
\langle \textit{NOTSituationRule} \rangle ::= & \textit{ NOT } \langle \textit{SituationRule} \rangle
\end{aligned}$$

Example A.0.1 *Nice Weather = { Warm AND Sunny }* shows an expression with a logical operator.

Context Free Grammars contain either terminal or non-terminal rules. Non-terminal rules contain other rules, whereas terminal rules are rules that match a token presented as a regular expression. All rules eventually end up with terminals. All leafs in the expression tree are terminals.

As mentioned before, a situation could depend on other situations. In order to represent that in the Situation Expression Language, situations are identified with their names, and consequently are referenced by them. *SituationToken* is an identifier representing a situation name.

$$\langle \textit{SituationToken} \rangle ::= \textit{ Identifier}$$

Example A.0.2 *Go Out = { NOT Stay Home }*, where *Stay Home* is a *SituationToken*.

Literal Situation is the mechanism used to define a dimension expression inside a Situation expression. A situation can be a simple situation which contains only dimension expressions in its definition. For example, *Hot Weather* situation is defined as { (Temperature > 30) }. This means that *Hot Weather* is a simple situation. Literal Situations can be used to define anonymous situations in the body of other situations. In Example A.0.3, { (Role == 'Admin') } is a *Literal Situation*, however that could be presented differently by explicitly defining a *Role Situation* and then referencing it in the *Admin* definition.

Example A.0.3 *Admin = { Authenticated AND { (Role == 'Admin') } }*

$$\langle \mathbf{LiteralExpression} \rangle ::= \{ \langle Dimension \rangle \}$$

Dimension Root represents the operations on context dimensions. These operations evaluate to either true or false, which determines if a situation exists in a given context or not.

$$\langle \mathbf{Dimension} \rangle ::= (\langle DimensionRule \rangle)$$

The purpose of Dimension rule is to make sure that each dimension expression starts with circle braces $()$ which eliminates any ambiguity when parsing these roles.

The dimension rules contains all the following operations.

1. Logical Operations: *AND, OR, NOT, Equal, NOT Equal, Bigger, Smaller, Bigger or Equal, Smaller, and Equal.*
2. Mathematical Operations: *Addition, Multiplication, Division, and Subtraction.*
3. User defined functions: They contain domain specific logic constructed to perform operations over dimensions. For example, users can define a function to check if a value is prime.

$$\begin{aligned}
\langle DimensionRule \rangle &:: \langle BraceDimension \rangle | \\
&\langle ANDDimensionRule \rangle | \\
&\langle ORDimensionRule \rangle | \\
&\langle FUNCDimensionRule \rangle | \\
&\langle NOTDimensionRule \rangle | \\
&\langle ADDDimensionRule \rangle | \\
&\langle DIVDimensionRule \rangle | \\
&\langle SUBDimensionRule \rangle | \\
&\langle MULDimensionRule \rangle | \\
&\langle EqualDimensionRule \rangle | \\
&\langle NotEqualDimensionRule \rangle | \\
&\langle BiggerDimensionRule \rangle | \\
&\langle BiggerOrEqualDimensionRule \rangle | \\
&\langle SmallerDimensionRule \rangle | \\
&\langle SmallerOrEqualDimensionRule \rangle | \\
&\langle TokenDimensionRule \rangle | \\
&\langle DimensionValue \rangle
\end{aligned}$$

The situation expression language includes logical dimensions operations. Logical operators at the dimension level are richer than their counterpart at the Situation level. In addition to AND, OR & NOT, compare operators can be used over dimensions since they have concrete values. Following is an example of the operations definitions.

$$\begin{aligned}
\langle ANDDimensionRule \rangle &:: \langle DimensionRule \rangle \\
&AND \langle DimensionRule \rangle \\
\langle BiggerOrEqualDimensionRule \rangle &:: \langle DimensionRule \rangle \\
&'>=' \langle DimensionRule \rangle
\end{aligned}$$

The situation expression language includes mathematical dimension operations. Mathematical operations are used over dimension values or the values supplied by users in the expression. All the four basic operations

are supported.

$$\langle \text{DIV DimensionRule} \rangle ::= \langle \text{DimensionRule} \rangle \text{"/"} \langle \text{DimensionRule} \rangle$$

Example A.0.4 *Hot Celsius* = { (Fahrenheit Temperature - 32) > 30) }

The situation expression language provides an extension point to bring user supplied logic to the reasoning operation. User defined functions provide a mechanism to extend the situation expression language. Functions should have one or more parameters which are either dimension values (tags) or user supplied values. Functions should return a Value. Syntactically, functions should start with an **\$** sign and the parameters should be in square braces []. Users should provide the implementation of the functions in Dynamic Link Libraries (DLL) that should be deployed in a special directory. The system is able to allocate their implementations at the run time, without recompilation. The following example illustrates user defined functions.

Example A.0.5 *Even Number* = { (**\$EvenNumber**[56, Dimension Name]) }

The situation expression language includes dimension terminals. Terminals, in case of dimensions, are either dimension name (identifier) or a dimension value (number or string). The following example shows a situation expression with the dimension name Temperature.

$$\langle \text{DimensionValue} \rangle ::= \langle \text{NUMBER} \rangle \mid \langle \text{STRING} \rangle$$

Example A.0.6 *Freezing* = { (Temperature =< 0) }

Appendix B

Workflow and Policy Expression Language

Adaptations contain a workflow of reactions that should be executed with respect to a set of policies. The *Workflow and Policy Expression Language* is used to define the adaptations. This language supports the following operations.

- Triggering reactions.
- Checking policies.
- Logical Operations (*AND*, *OR*, & *NOT*). and
- Control constructs (*WHILE*, *IF ELSE*, and *FOR*) to allow rich workflow expressions.

The workflow and policy expression language is defined using context free grammar as follows.

The root rule for the language is the Workflow rule. The workflow is simply a statement collection.

$$\langle \textit{Workflow} \rangle ::= \langle \textit{StatementCollection} \rangle$$

The statement collection is a recursive rule (star rule) over the statement rule. Which means that each adaptation can contain one statement or more.

$$\begin{aligned} \langle \textit{StatementCollection} \rangle ::= \langle \textit{StatementCollection} \rangle \langle \textit{Statement} \rangle \mid \\ \langle \textit{Statement} \rangle \end{aligned}$$

The Statement rule is the main bulk of the workflow language. The Workflow Expression Language supports the following statements.

- *IF ELSE* Statement,
- *While* Statement,
- *For* Statement,
- *Execute* Statement and
- *Brace Statement*: It represents that a statement can be encapsulated in braces to enforce operator precedence.

The definition of the Statement rule is presented as follows.

$$\begin{aligned} \langle \textit{Statement} \rangle ::= & \langle \textit{WhileStatement} \rangle | \\ & \langle \textit{ForStatement} \rangle | \\ & \langle \textit{IfElseStatement} \rangle | \\ & \langle \textit{ExecuteStatement} \rangle | \\ & \langle \textit{BraceStatement} \rangle \end{aligned}$$

The Condition Statement is used as a part of the *IF Statement* and *While Statement*. Conditions are either a policy check or a logical expression over other conditions.

$$\begin{aligned} \langle \textit{Condition} \rangle ::= & \langle \textit{Condition} \rangle | \\ & \langle \textit{ANDCondition} \rangle | \\ & \langle \textit{ORCondition} \rangle | \\ & \langle \textit{NOTCondition} \rangle | \\ & \langle \textit{PolicyCheck} \rangle \end{aligned}$$

The logical condition statement is a logical aggregation of other statements. The following shows the defini-

tions of these rules.

$$\begin{aligned} \langle \text{ANDCondition} \rangle &::= \langle \text{Condition} \rangle \text{ AND } \langle \text{Condition} \rangle \\ \langle \text{ORCondition} \rangle &::= \langle \text{Condition} \rangle \text{ OR } \langle \text{Condition} \rangle \\ \langle \text{NotCondition} \rangle &::= \text{NOT } \langle \text{Condition} \rangle \end{aligned}$$

Example B.0.1 $\$IsAuthorized[\text{User ID}] \text{ AND } \$IsInRole[\text{User ID}]$

The policy check statement is the mechanism used to check execution policies in the workflow. Each policy check statement is a call to a user defined function that accepts zero or more parameters. This function returns a Boolean value to indicate whether the policy has evaluated to true or false. Policy Name is a Terminal rule that matches string identifier which starts with an \$ sign to eliminate ambiguity when parsing the language.

$$\begin{aligned} \langle \text{PolicyCheck} \rangle &::= \langle \text{PolicyName} \rangle [\langle \text{ParamList} \rangle] \\ &\langle \text{PolicyName} \rangle [\] \end{aligned}$$

Example B.0.2 $\$IsAuthorized[\text{User ID}]$

The *While Statement* is used to repeatedly execute a set of actions as long as a condition is met. The While Statement contains two parts (1) a condition, and (2) a body which is a statement.

$$\langle \text{WhileStatement} \rangle ::= \text{while } \langle \text{Condition} \rangle \langle \text{Statement} \rangle$$

The For Statement is used to repeatedly execute a set of actions for a fixed number of times. The For Statement contains two parts (1) a number indicating the number of iterations, and (2) a body, which is a statement.

$$\langle \text{ForStatement} \rangle ::= \text{for}(\langle \text{NUMBER} \rangle) \langle \text{Statement} \rangle$$

The *If statement* is used to execute different branches based on a condition. The *IF Statement* is a typical example of a shift-reduce conflict, the preferred behavior in this case is the shift (whenever the parser

encounters an “else” the parser should shift rather than reduce with the IF rule).

$$\begin{aligned} \langle \textit{IfElseStatement} \rangle ::= & \langle \textit{IfStatement} \rangle \mid \\ & \langle \textit{IfStatement} \rangle \textit{ else } \langle \textit{Statement} \rangle \textit{ *preferred} \end{aligned}$$

$$\langle \textit{IfStatement} \rangle ::= \textit{ if } \langle \textit{Condition} \rangle \langle \textit{Statement} \rangle$$

The *Execute Statement* is used for triggering reactions. Reactions are user defined functions, and are identified with their names. Each reaction may have zero or more parameters that are either user supplied value or dimension context information. Reaction Name is a terminal rule that matches a string Identifier.

$$\langle \textit{ExecuteStatement} \rangle ::= \textit{ Exec } (\langle \textit{ReactionName} \rangle) ;$$

$$\begin{aligned} \langle \textit{Reaction} \rangle ::= & \langle \textit{ReactionName} \rangle [\langle \textit{ParamList} \rangle] \mid \\ & \langle \textit{ReactionName} \rangle [] \end{aligned}$$

The *Param List* is a plus closure rule that matches one parameter or more, each parameter is either a string value in single quotation, or an integer value.

$$\begin{aligned} \langle \textit{ParamList} \rangle ::= & \langle \textit{ParamList} \rangle , \langle \textit{Param} \rangle \mid \\ & \langle \textit{Param} \rangle \end{aligned}$$

Appendix C

Pseudo code

This section contains the pseudo code for the different functionalities of the modules in the architecture.

Algorithm 1 IContextManager

```
function COMPARESENSORS(iSensorReading, iSensor)
  if iSensorReading.id ≠ iSensor.id then
    return false
  end if
  if iSensorReading.type ≠ iSensor.type then
    return false
  end if
  if iSensorReading.unit ≠ iSensor.unit then
    return false
  end if
  return true
end function
function VALIDATESENSORREADING(iSensorReading)
  if (iSensorReading.time − Time.Now) > 15 then
    return Null
  end if
  if compareSensors(iSensorReading, sensor) == false then
    return Null
  end if
  return validateRange(iSensorReading)
end function
```

Algorithm 2 ISituationManager

```
function GETPATIENTSITUATIONS(iContext)
  patientSituations ← []
  j ← 0
  for i = 0 to situations.length - 1 do
    if situations[i].id == iContext.patientId then
      patientSituations[j] ← situations[i]
      j ← j + 1
    end if
  end for
  return patientSituations
end function
function GETCONTEXTSITUATIONS(iContext)
  patientSituations ← getPatientSituations(iContext)
  contextSituations ← []
  j ← 0
  for i = 0 to patientSituations.length - 1 do
    if patientSituations[i].type ≠ iContext.type then
      continue
    end if
    if patientSituations[i].unit ≠ iContext.unit then
      continue
    end if
    contextSituations[j] ← patientSituations[i]
    j ← j + 1
  end for
  return contextSituations
end function
```

Algorithm 3 IAdaptationManager

```
function FETCHSITUATIONADAPTATIONS(situations)
  situationAdaptations ← []
  k ← 0
  for i = 0 to situations.length - 1 do
    for j = 0 to healthModel.adaptations.length - 1 do
      if healthModel.adaptations[j].condition ≠ situations[i].condition then
        continue
      end if
      if healthModel.adaptations[j].situation ≠ situations[i].name then
        continue
      end if
      situationAdaptations[k] ← healthModel.adaptations[j]
      k ← k + 1
    end for
  end for
  return situationAdaptations
end function
```
