Formal description of the Home UbiHealth model using Denotational Mathematics

DISSERTATION

Submitted in fulfillment of the requirements
for the degree of
Doctor of Philosophy in Computer Science
Department of Computer Science and Engineering
University of Ioannina

by

Ioannis Constantine Sarivougioukas

October 2016

Τριμελής Συμβουλευτική Επιτροπή

- Ισαάκ Λαγαρής, Καθηγητής, Τμήμα Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο Ιωαννίνων (Επιβλέπων).

- Λυσίμαχος-Παύλος Κόντης, Αναπληρωτής Καθηγητής, Τμήμα Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο Ιωαννίνων.

- Κωνσταντίνος Παρσόπουλος, Αναπληρωτής Καθηγητής, Τμήμα Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο Ιωαννίνων.

Επταμελής Εξεταστική Επιτροπή

- Ισαάκ Λαγαρής, Καθηγητής, Τμήμα Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο Ιωαννίνων (Πρόεδρος).

- Λυσίμαχος-Παύλος Κόντης, Αναπληρωτής Καθηγητής, Τμήμα Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο Ιωαννίνων.

- Κωνσταντίνος Παρσόπουλος, Αναπληρωτής Καθηγητής, Τμήμα Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο Ιωαννίνων.

- Κωνσταντίνος Μπλέκας, Αναπληρωτής Καθηγητής, Τμήμα Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο Ιωαννίνων.

- Γεώργιος Μαντής, Επίκουρος Καθηγητής, Τμήμα Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο Ιωαννίνων.

- Δημήτριος Φωτιάδης, Καθηγητής, Τμήμα Μηχανικών Επιστήμης Υλικών, Πανεπιστήμιο Ιωαννίνων.

- Διονύσιος-Δημήτριος Κουτσούρης, Καθηγητής, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Εθνικό Μετσόβιο Πολυτεχνείο.

ACKNOWLEDGEMENTS

“I am indebted to my father for living, but to my teacher for living well.”
(His teacher was the legendary philosopher Aristotle)
— Alexander The Great

I would like to express my gratitude to Professor Isaac Lagaris for the provision of mentoring throughout the preparation course of this research work.

I thank Professor Constantine Parsopoulos for advising and providing the freedom to navigate and explore challenging scientific research horizons with the tirelessly and wise support to be always on tracks.

Also, I would like to thank those believed in me supporting my efforts.

CONTENTS

TABLES INDEX

FIGURES INDEX

EXECUTIVE SUMMURY

Ioannis Sarivougioukas, Constantinos and Orsalia, Ph.D., Computer Science Department, University of Ioannina, September, 2015, “ Formal description of the Home UbiHealth model using Denotational Mathematics”, Mentor: Dr. Isaac Lagaris.

Science and engineering support Healthcare contributing in well-being and prosperity. The scientific attainments complement the technological achievements of nano-technology, communications, and computing developing sustainable paradigms.

The discrete levels of Health Systems are challenged by the scarcity of medically controllable facilities. The tertiary and secondary levels of the Health Systems strive to provide extremely specialized and costly hospitals. At the primary level, distinguished into formal and informal, the limited availability of health centers and medical offices of formal Primary Care raises issues related with medical risks, funding, and equity in access. The informal Primary Care refers to the provision of healthcare services at medically uncontrollable environments such as at homes.

Research efforts for the provision of medical services at home, Home Healthcare, confronted by limited standardization in the domestic environment and adopted the use of technological paradigms such as Smart Home, Internet of Things, Cloud Computing and Ubiquitous Computing along with the medical paradigm of Medical Home in Primary Care. The challenges refer to make the informal Primary Care medically controllable capable to substitute specific accountabilities of the rest of the levels of the Health System reconfiguring the complex flows of information within the house and among Government Agencies, Health Authorities, Social Security, Insurances, and the Market.

Home can be medically administrable with support from ubiquitous computing, adopting principles of Smart Home, relying on Internet of Things, and using Cloud Computing for its systemic integration. Home healthcare in ubiquitous computing environments, Home UbiHealth, must incorporate software with principles of autonomy, mobility, migration, and polymorphism.

The complex healthcare processes, inside and outside of the house, requires the design and implementation of complicated software applications. Analytical mathematics have limited expressiveness along with the disadvantage of large volumes of tedious analyses and methods due to extensive calculations. Moreover, algebra and calculus are incapable to manipulate medical concepts, and formally describe the dynamic behavior of software applications.

The contribution of this research work is the capability to describe rigorously and formally with Denotational Mathematics the Home UbiHealth model. In particular, this research work provides a tool set in a formally given framework for the description of the behavior of software ahead of its implementation.

EXTENDED ABSTRACT

Ιωάννης Σαριβουγιούκας, του Κωνσταντίνου και της Ορσαλίας, Ph.D., Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Ιωαννίνων, Ιούνιος, 2016, "Τυπική περιγραφή του μοντέλου Home UbiHealth με χρήση Δηλωτικών Μαθηματικών", Επιβλέπων Καθηγητής: Ισαάκ Λαγαρής.

Το περιβάλλον του σπιτιού καθίσταται εχθρικό για τις τεχνολογίες Πληροφορικής και βιοϊατρικών εφαρμογών λόγω της έλλειψης τυποποίησης και των ανεξέλεγκτων συνθηκών που επικρατούν. Επίσης, η τυπική περιγραφή συστημάτων και παρεχόμενων υπηρεσιών Υγείας με αναλυτικά μαθηματικά απαιτεί εκτεταμένες, περίπλοκες και εξαντλητικά λεπτομερείς περιγραφές. Για παράδειγμα, ο ρυθμός έκχυσης ινσουλίνης μιας εμφυτευμένης αντλίας μπορεί να υπολογιστεί με ακρίβεια και να περιγραφεί το σύστημα τυπικά χρησιμοποιώντας διαφορικές εξισώσεις, απαιτώντας πλήθος πολύπλοκων αλγεβρικών πράξεων και εκτενή εκπαίδευση. Εν τούτοις, οι αναλυτικές μαθηματικές μέθοδοι παρέχουν μεν ακριβή στοιχεία αλλά χρειάζεται να καταβληθεί επιπλέον προσπάθεια με μαθηματικές μεθόδους για την λήψη πληροφοριών που σχετίζονται με την λειτουργικότητα και την συμπεριφορά του συστήματος [1].

Για την έκφραση και την επικοινωνία μεταξύ των επιμέρους αναλυτικών μεθόδων των αναλυτικών μαθηματικών, απαιτείται η καταβολή πρόσθετης προσπάθειας με την χρήση κατάλληλων μαθηματικών μεθόδων, όπως συμβαίνει με την μεταφορά των αποτελεσμάτων επίλυσης διαφορικών εξισώσεων του παραδείγματος σε άλλους μαθηματικούς τομείς όπως η Λογική, ο π-Λογισμός, η Θεωρία Κατηγοριών και τα δίκτυα Petri. Επιπλέον, οι ιδιότητες και τα χαρακτηριστικά του Διάχυτου Υπολογισμού μπορούν να υποστηρίξουν την παροχή υπηρεσιών Υγείας στο Σπίτι ενώ υιοθετώντας κι άλλες τεχνοτροπίες όπως το Διαδίκτυο των Πραγμάτων και ο Υπολογισμός στο Σύννεφο μπορούν να συγκροτήσουν νέα μοντέλα και νέες πρακτικές μεθόδους που περιγράφονται με τον όρο Home UbiHealth.

Σταθεροί, φορητοί και εμφυτευμένοι αισθητήρες και μηχανισμοί ενεργοποίησης συσκευών και βιοιατρικών διατάξεων συγκροτούν υποσυστήματα τα οποία μπορούν να εξυπηρετήσουν εξατομικευμένους ιατρικούς σκοπούς στο περιβάλλον του Διάχυτου Υπολογισμού, λαμβάνοντας υπόψη τις προτιμήσεις του υποστηριζόμενου ασθενή. Οι διάσπαρτες συσκευές στο Σπίτι συνεργάζονται είτε επικοινωνώντας απευθείας μεταξύ τους ή μέσω ενδιάμεσων συστημάτων εξαγωγής συμπερασμάτων που επιτρέπουν την συλλειτουργία συστημάτων μέσω της σημειολογικής και εννοιολογική τους συνεργασίας. Για την περιγραφή της συλλειτουργίας, απαιτείται ο ορισμός ενός τυπικού μαθηματικού πλαισίου για την διαχείριση εννοιολογικών διεργασιών και σημειολογικών εννοιών που έχουν ως σκοπό την στοχευόμενη λήψη αποφάσεων για την καθοδήγηση ανεξάρτητων φυσικών συσκευών.

Για τον μετασχηματισμό σημάτων και πρωτογενών στοιχείων σε ορθές και πλήρεις πληροφορίες απαιτείται μια δομή τέτοια ώστε να μπορεί να επεξεργάζεται τα παραλαμβανόμενα δεδομένα συνθέτοντας και αναπτύσσοντας πληροφορίες. Στην συνέχεια, οι πληροφορίες μετατρέπονται και συγκροτούν δίκτυα συσχετιζόμενων εννοιών, δεδομένης της υποστήριξης από ενδεδειγμένη γνώση ενήμερων ιατρικών ταξινομιών και λαμβάνοντας υπόψη τους εξατομικευμένους περιορισμούς που αφορούν τον ασθενή με κατάλληλες διεπαφές. Η διάδραση μεταξύ των κόμβων του συγκροτούμενου δικτύου εννοιών ελέγχεται από σύστημα ελέγχου το οποίο διαχειρίζεται την επικοινωνία μεταξύ των εννοιών και την τελική διαμόρφωση του εν λόγω δικτύου ώστε να επιτυγχάνονται κάθε φορά οι τιθέμενοι ιατρικοί σκοποί και στόχοι υπό το πρίσμα των εξατομικευμένων ιατρικών αναγκών του ασθενή. Το σύστημα ελέγχου λαμβάνει αποφάσεις και μεταφέρει τα αποτελέσματα της διάδρασης μεταξύ των εννοιών στις διεπαφές των συσκευών που υποστηρίζουν τον ασθενή. Η φυσική δραστηριότητα γίνεται καταληπτή σε κατάλληλα χρονικά διαστήματα και προκαλείται επεξεργασία η οποία επιστρέφει τις τιμές των παραμέτρων των εξαγόμενων συμπερασμάτων στις συσκευές που συλλειτουργούν στο φυσικό περιβάλλον του Σπιτιού.

Οι διεργασίες που λαμβάνουν χώρα και εξελίσσονται στο περιβάλλον του Home UbiHealth περιορίζονται από τα εφαρμοζόμενα εξατομικευμένα ιατρικά πρωτόκολλα και τις οδηγίες για την θεραπεία του ασθενή. Η διαδικασία μετατροπής δεδομένων σε έννοιες και η περαιτέρω διαχείριση των εκτελούμενων εργασιών πραγματοποιείται εφαρμόζοντας Άλγεβρα Διεργασιών [2] επί των συγκροτούμενων μετα-διεργασιών ενώ η διαχείριση των εννοιών πραγματοποιείται με την Άλγεβρα Εννοιών [3] εφαρμοζόμενη επί δικτύου εννοιών. Η αλγεβρική διαχείριση προκαλεί την διαμόρφωση εσωτερικών υποσυστημάτων τα οποία ελέγχονται και διαχειρίζονται με την συνδρομή εργαλείων από την Θεωρία Ελέγχου όπως για παράδειγμα, της ελεγξιμότητας, της παρατηρησιμότητας, της ευστάθειας, σταθερότητας, συντηρησιμότητας, προσβασιμότητας και της ανιχνευσιμότητας για την υλοποίηση εξατομικευμένων ιατρικών θεραπευτικών προσεγγίσεων.

Αυξανόμενου του αριθμού συσκευών που υποστηρίζουν τον ασθενή στο Σπίτι, αυξάνει αναλογικά ο χρόνος υπολογιστικής επεξεργασίας και συνεπώς, η απόκριση του συστήματος φθίνει απαιτώντας ρυθμιστικές παρεμβάσεις. Για την ανταπόκριση στο φόρτο επεξεργασίας, προσαρμόζεται το επίπεδο διακριτότητας των εννοιών υποστηριζόμενο από κατάλληλα μοντέλα γνώσης και νόησης. Μ’ άλλα λόγια, εάν το επίπεδο διακριτότητας των εννοιών ενός δικτύου εννοιών προκαλεί απόκριση εκτός των προκαθορισμένων ορίων απόδοσης τότε η διακριτότητα μπορεί να μειωθεί ώστε να ασκηθεί έλεγχος στο δίκτυο εννοιών με λιγότερα χαρακτηριστικά. Αντίθετα, σε περιπτώσεις εννοιολογικών αντιθέσεων, η διακριτότητα μπορεί να αυξηθεί ώστε να αναχθεί το πρόβλημα σε χαμηλότερο επίπεδο εννοιών για την ανάδειξη και εν τέλει την ρύθμιση των παραγόντων που προκαλούν αντιθέσεις.

Τα Δηλωτικά Μαθηματικά ορίζουν το πλαίσιο για την τυπική περιγραφή της στατικής και δυναμικής συμπεριφοράς του μοντέλου νοσηλείας στο Σπίτι σε περιβάλλον Διάχυτου Υπολογισμού ή Home UbiHealth. Το πλαίσιο διαχειρίζεται έννοιες που αλληλεπιδρούν για να επιτύχουν εκείνες τις συσχετίσεις που επιτρέπουν την ιατρική και νοσηλευτική εξατομίκευση των υπηρεσιών στο Σπίτι. Αξιοποιώντας κατάλληλες διεπαφές, οι επαγγελματίες Υγείας μπορούν να επικοινωνήσουν, τοπικά και από μακριά, με το εν λόγω σύστημα υποστήριξης στο Σπίτι χρησιμοποιώντας νοητικές προσεγγίσεις χωρίς την υποχρέωση προσδιορισμού τιμών σε μεταβλητές και αξιοποιώντας την αναπτυσσόμενη νόηση.

CHAPTER 1. INTRODUCTION

UbiComp (Ubiquitous Computing) is the term introduced by Mark Weiser in 1988 [1.1] to describe an alternative human-computer interaction. UbiComp refers to the post-desktop period where data is integrated within the surrounding devices-objects when carrying out the everyday activities. The use of the surrounding devices is performed unconsciously[1.2], without paying attention to the use of the device, the same way some is using the eye-glasses, focusing on the performing task instead of the use of the employed tool. For this reason, the UbiComp requires the readiness of the surrounding devices to be available anywhere and at any time [1.3]. The unconscious use of the surrounding devices allows them to enter the user’s world [1.4] because the devices operate consciously striving to serve the user in an intelligent manner, instead of following the virtual reality paradigm where the user enters the world of the computing devices. The employed computing devices can be classified with respect to their mobility property as stationary, portable, wearable, and implantable [1.5]. The availability of computing devices and sensors stimulated research initiatives such as the Smart Home paradigm [1.6] and Telemedicine applications requiring the property of sustainability [1.7]. As illustrated at Figure-1.1,

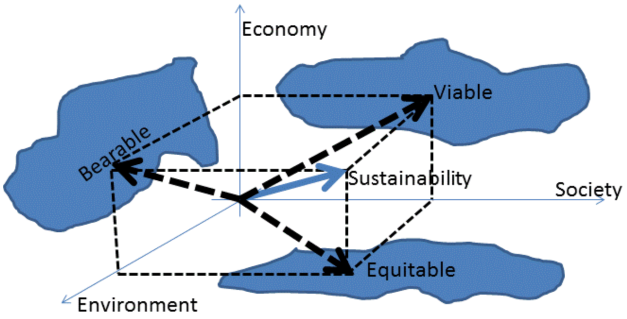


Figure-1.1 Sustainability features of ubiquitous computing.

Home	Role	Stakeholder	Benefits
Inside	Patient	Person or individual	Treatment at home, personal preferences
	Medical Professionals	Doctors Nurses Caregivers & Therapists	Evidence acquisition - Improved collaboration, communication and documentation - Scheduling and treatment plans
	Supporters	Family Relatives Friends Volunteers	Active assistive participation
Outside	Society	Community Social Security Agencies Consumers unions Payers	Awareness - Promptly support and reduced costs - Quality of services Justifiable expenses, reduced costs
	Authorities	Government Health Authorities Hospitals, Health Centers Emergency Departments	Healthcare policing and spending - Epidemiological monitoring - Reduced re-hospitalizations - Readiness
	Industry	Biomedicals Pharmaceuticals Medical Products & Business Suppliers	Patients’ needs satisfaction - Reporting Quality - Reduced administrative costs

Table-1.1 Stakeholders Benefits.

these applications require the property of sustainability [1.8] due to their significant impacts on economic activity, society, and healthcare in ordinary everyday life. Successful research efforts achieved the provision of monitoring services at home [1.9] following particular parameters such as vital signs or glucose level in blood, and monitoring aspects of the daily activity.

UbiComp facilitated healthcare for the development of e-Health which evolved into m-Health with the proper adoption of mobile devices and software applications [1.10]. An alternative approach considers the use of Smart Home paradigm [1.11] from the Healthcare point of view. The success of any proposed

Healthcare model [1.12] must present stable equilibrium among its constituting components, i.e., the Environment, the Society, and the Economy. The Environment represents the ecosystem of typical a house and its relations with external medical professionals, the involved Authorities, and the members of the Market [1.13]. Economy represents the infrastructure of money flows between the healthcare suppliers and consumers of products and services [1.14]. The Society factor of sustainability refers to all members of Society participating in the provision of healthcare at home [1.15]. Table-1.1 presents an indicative list of stakeholders according to their position with respect to the house. Hence, the development of a model must obey the sustainability prerequisites in order to achieve its mission.

1.1 From e-Health to m-Health

The introduction of Information and Communication Technology (ICT) led to the development of Enterprise Resource Planning (ERP) systems [1.16] applied on closed and controllable medical environments. The provided electronic healthcare (eHealth) services spanning from e-scheduling, e-prescriptions, e-payments, to electronic medical records (EMR), aim at the efficient management of the level of quality [1.17] of medical services, the reduction of the involved costs, and the patients empowerment administering their own information. However, e-Health inherits the complexity of healthcare [1.18] processes originated from multiple sources involving geographical, educational, cultural, religious, social, professional, economical, and political factors that directly affect diseases and burdens morbidity and mortality rates.

e-Health implementations are missing widely acceptable standards [1.19], e.g., codifications in hospitals, limiting the continuity of medical information flows. The continuity of information flows among ERPs forms network-centric healthcare services developing three domains [1.20]: the physical, the information, and the cognitive domain. The physical domain refers to the environment of healthcare services affected by the physical means and factors such as the personnel's training, the patients' culture, the financing, and the political environment. The information domain refers to the cycle of information's life, from spawning to dissemination and final storage, as well as the development of knowledge from processing related information. The cognitive domain refers to the interrelation of the realized human factors in the execution of the related activities such as organization, experience, motivation, and intuition. Therefore, e-Health requires continuity in the flows of information [1.20] within compatible physical, information, and cognitive domains.

Wireless communications extended the desktop paradigm of e-Health to mobile healthcare (mHealth) services. Mobile computing and communication devices adopt wireless communication protocols, e.g., the global system for mobile (GSM), forming hierarchical networks based on the proximity connection distance accomplishing real-time processing of m-Health services [1.21]. The developed m-Health systems can be classified [1.22] according their use into (i) administrative tasks such as m-scheduling and m-prescriptions, (ii) supporting tasks occurring on user's behalf such as e-billing, and (iii) medical tasks achieving the interconnection of the patient with the healthcare providers such as m-diagnostics or m-monitoring. The m-Health applications present the characteristics [1.23] of the immediacy in emergency cases, interactive remote consultation, patients' empowerment, personalization and continuous health monitoring with personalized adapted sensors. The spread of m-Health is limited by the lack of m-protocols that achieve the development of m-services integrated or connected with e-Health systems, the associated communication costs, the scarcity and the cost of medical professionals for setting up real-time consultation sessions, the payment methods for the provided m-Health professional services, and the incapability to present substantial positive economic impact [1.24].

1.2 Smart Home

Smart Home refers to the paradigm of interconnecting electrical home appliances and the associated services can be controlled, monitored, and accessed even remotely [1.25]. The operation is performed without interventions making decisions based on the developed, each time, context, applying predetermined logic and invisible computing [1.26]. The model directly addresses issues related to security, entertainment, and home environment while the healthcare maters emerged as a natural consequence [1.27]. The reduction of the production cost of small in scale devices that can be embedded in many objects making them almost invisible and operating without connecting wires developed the necessary prerequisites for the first implementations of the model. The massive production of low cost products such as sensors, proximity cards, radio-frequency identification (RFID) artifacts and other smart devices of reduced size incorporating with computing devices supported by adequate software deployments provide the capability to function according to the developed context. Hence, software applications incorporate and synchronize the interconnected smart devices [1.28].

The ubiquitous computing paradigm in the development of the Smart Home model raises technical and social issues [1.29] related with the following issues:

- *Use:* the operation is achieved with minimal habitants' interventions due to smart devices capabilities to learn from the residents' behavior through sensors such as recognizing gestures or voices. However, the existing houses were not designed to facilitate such systems as the initiative of Aware Home at Georgia Tech.
- *Interoperability:* appliances from different manufacturers are called to participate obtaining the desired functionality in an unplanned interoperability scheme, controlled by a centralized local computing service without using user-interfaces.
- *Administration:* the model self-recovers from failures activating processes and initiating procedures for assistance on behalf of the habitant. The system performs software updates and raises notifications to the user concerning hardware failures.
- *Reliability:* the implementations never experience crash-conditions and the design of the model's control must be modular prohibiting the failure of a subsystem to affect the rest of the system. The introduced autonomy must anticipate the system from false inputs sensing multiple signals protecting the network against attacks due to critical decisions concerning the habitants' safety.
- *Inference and ambiguity:* mechanical failures, interference, and white nose oblige the system to make decisions on low quality data. Control schemes track the expecting behaviors providing the availability to users to override made decisions.
- *Adaptability:* the software applications must adapt to the dynamically evolving conditions at home by changing frequently targets and aims.

The availability of wirelessly connected sensors and actuators supported by databases and software applications permits the extension of the developed model over the internet achieving cloud-based Smart-Home implementations [1.30].

HEALTH STATE		POPULATION	HEALTHCARE LEVEL
1	Healty	Largest portion of all ages	Primary
2	Crisis	Smallest portion	Secondary, Tertiary
3	Chronic	Largest portion among the aged including the handicapped and the impaired	Primary, Secondary or/and Tertiary

Table-1.2 States of Health.

1.3 Home Healthcare

Healthcare is concerned with the health states of the healthy, those facing some crisis, or live with some, temporary or permanent, diagnosed disease or disability, as reported in Table-1.2. Home healthcare is concerned with all three classes to obtain the best possible status [1.31]. The Canadian Institute for Health Information in 1999 published [1.32] the following definition about Home Care:

> “A range of health-related, social and support services received at home. These services enable clients incapacitated, in whole or in part, to live in their home environment. These services help individuals achieve and maintain optimal health, well-being and functional ability through a process of assessment, case co-ordination, and/or the provision of services. Service recipients may have one or more chronic health conditions or recently experienced an acute episode of illness or hospitalization. The range of services provided includes prevention, maintenance, rehabilitation, support and palliation.”.

The American Academy of Pediatrics in 1992 published guidelines [1.33] for the development of the Medical Home model. According to the issued guidelines the primary care physicians urged to include in their practice (a) the promotion of prevention in the provided care, (b) the assertion of the availability of inpatient care, (c) the availability of continuity of care at all ages, (d) the availability of referrals and consultation from various medical specialties, (e) the capability to communicate with local schools and community authorities, (f) the development of files with all related and relevant important medical data and information [1.33]. The Medical Home model is the first formal approach to establish Home Healthcare services while the medical professionals face objective problems implementing various versions of it [1.34].

The importance of the Medical Home model urged the collaboration of the American Academy of Pediatrics, the American Academy of Family Physicians, and the American College of Physicians to issue a Clinical Report on Transition for the transfer of medical responsibility from pediatric to adult care [1.35]. However, there is still missing, to the best of my knowledge, a systemic, standardized, and rigorously defined formal model for Home Care.

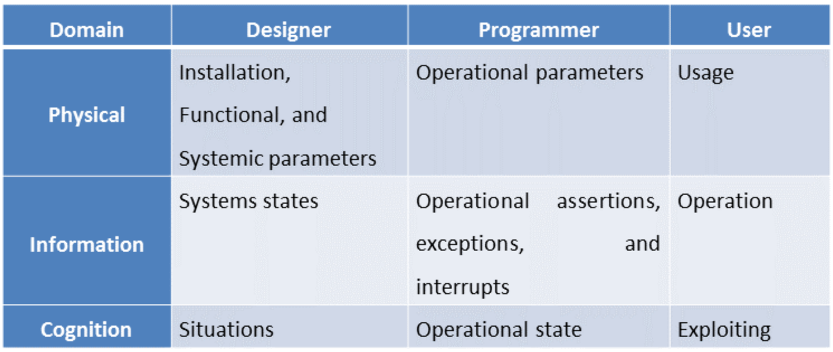
1.4 Formal mathematical expressiveness: Denotational Mathematics

The formal mathematical description of the involved informatics infrastructures, operations, and functionality of Healthcare services at the home environment requires more expressive mathematical means than those provided by analytical mathematics, numbers and set theory [1.36]. The home environment is missing standardization making it hostile for Information Technology healthcare applications requiring high-level composite and attributed mathematical constructs [1.36] to describe formally the necessary software infrastructure along with its static properties and its dynamic behavior. Home healthcare in an ubiquitous computing environment requires the aid from Cognitive Informatics, which is an inter-disciplinary field receiving contributions from computer science, information science, and cognitive science for the use and exploiting of advanced mechanisms processing information and concepts with engineering methods resulting into Cognitive Computing. Moreover, Cognitive Informatics [1.36] requires (i) the capability to administer natural intelligence, (ii) the ability to develop abstractions of intelligence, (iii) the capability to express formally mathematical models with Denotational Mathematics, and (iv) the feasibility to perform cognitive processing of intelligence with Cognitive Computing. Those four requirements constitute the frame of Cognitive Computing applications processing knowledge by a computer model, analogous to von Neumann’s, consisted of a Cognitive central processing unit (CPU) performing inferences on conceptual abstractions formally represented by Denotational Mathematics.

Denotational Mathematics provides a formal theoretical framework to manipulate objects’ abstractions with complex relations conveying perceptions of conceptual abstractions and knowledge [1.37]. This mathematical framework allows to formally describing the static and dynamic intelligent behaviors of the processes represented by operating software procedures along with the capability to apply Systems Theory to manipulate the administered entities. The formally defined tools in the Denotational Mathematics theoretical framework are capable of manipulating concepts with Concept Algebra, well-defined systems with the tool of Systems Algebra, formally determined constructs with Granular Algebra, specifically designated operations with Real-Time Process Algebra, and formally interpreted causalities with Inference Algebra [1.37]. The expressiveness of the Denotational Mathematics tools set feasible the attempt to formally describe the infrastructure, the static properties, and the dynamic behavior of complex software systems. Among the developed models, such as the Layered Reference Model of the Brain (LRMB) [1.38], the Object-Attribute-Relation (OAR) model [1.39], the models of the brain and the neuro-informatics models [1.40], provide the opportunity to model a system that performs cognitive operations analogous to the modeled brain functions processing medical knowledge achieving the cognitive interaction of the system with the medical professionals and the attended individuals too.

1.5 Benefits and uses of Denotational Mathematics

Denotational Mathematics formally describes the static and dynamic behaviors of software systems that implement workflows of activities aiming to assist the designer, the programmer, and the interacting end-users [1.41]. For each activity, Denotational Mathematics formally describes the supporting data infrastructures such as variables, data structures, and objects providing the static behavior of the activity. Moreover, Denotational Mathematics provides the means to formally describe the dynamic behavior of the performing software providing efficiently the data infrastructure's interdependencies and functionality. The formal description addresses the needs of the activities' designer [1.41] by providing the means to control the efficiency, appropriateness, and effectiveness of the involved activity, examining the discrete objectives and the limits of operation of the formed systems. The programmer can be assisted by controlling the aims [1.41] of specific components, examining, managing, and verifying the raised exceptions, and following the communication procedures among the systems' components. The end-users enjoy the capabilities provided by Denotational Mathematics to manage systems of concepts of varying



Domain	Designer	Programmer	User
Physical	Installation, Functional, and Systemic parameters	Operational parameters	Usage
Information	Systems states	Operational assertions, exceptions, and interrupts	Operation
Cognition	Situations	Operational state	Exploiting

Table-1.3 Correlating domains and users roles.

granularity and entropy [1.41] achieving adapted near-natural interaction with the formed systems taking advantage of the available forms of knowledge.

As an example, consider the case of an implanted insulin fusing pump for diabetics at home. Denotational Mathematics provides the designer with the capability to control the static behavior involving the installation parameters of the designed application system, the parameters determining the acceptance of the pump by the patient’s organism, and the parameters defining the appropriateness of the pumps’ technical and operational characteristics. In addition, the designer is capable to control the dynamic characteristics which are the functional parameters determining the behavior, such as the sensors activating the pump functionalities, the efficiency and the functional response of the pump according to the specific needs of the particular patient, and the communication of the pump with the rest of the systems supporting the patient. The programmer is able to examine, verify, and validate the operation of the pump guided by software. The programmer needs to control the availability of the soundness and completeness of the necessary parameters such as the input, for instance the temperature, pressure, pulses, the level of glucose, and the level of insulin in its storage container. Also, the programmer can control the availability of the software functions that activate the associated mechanisms of the pump such as the rate of flow of insulin and the availability level of insulin. In addition, the programmer is able to examine individual and combinations of raised assertions and exemptions during the operation of the software, continuing the same example, raising interrupts and sending codified messages to the controlling unit of the pump or to the systems associated with the pump. Continuing the example, the end-user can be the medical professional that loads or inspects the pump, the specialized professional that adjusts its operation, the doctor that performs a clinical examination, the nurse in charge that cares about the availability of insulin, and the patient that experiences the daily operation of the pump.

Exploiting the availability of the Denotational Mathematics to formally describe systems' behaviors, a formally defined interface can be developed to manipulate the systems' contents of the corresponding physical, information, and cognitive domains [1.20]. Such an interface can adjust, each time, the level of entropy during the communication among users and systems according to the developing context. Continuing the above example, the interface's responses according to the kind of user and the domain within the evolving context is provided in Table-1.3 and hence, if the pulse rate is rapidly increasing, the arterial pressure is above normal, there is perspiration, breathlessness, fever, and the glucose level is out of the range of acceptable values, then there are activated the corresponding alarms at the physical domain of the system that supports the individual. At the information domain, if the vital signs and the level of glucose in the blood is out of physiological limits then there are activated the corresponding medical protocols and algorithms to confront the situation. If the clinical condition of the individual presents such evidences in the physical domain and such indications in the information domain, then it is possible an allergic shock in the cognitive domain, unless there are other reasons from other systems such as the respiratory, the cardiac, or the neurological systems of the individual's organization.

The capability of the system adapting its level of detail in the interaction with the users depends on its potentiality to respond each time at different domains and at adjusted level of conceptual granularity [1.42]. In the same example, at the physical domain the vital signs can be expressed analytically or aggregated using proper notation such as normal, high, low, high-high, or low-low alarms. At the information domain, the decisions taken by the system can refer at some crisis of the respiratory or the vascular systems in conjunction with diabetes. At the cognitive domain, the individual undergoes a shock which can be caused by the presence of the implanted pump and the physician can interact either with the details of the physical domain, or the information related to a specific event, or the involved concepts, in this case, the medical concept of the allergic shock, and the interaction with the parameters defining the allergic shock related to implantation of insulin infusion pump.

1.6 Motivation and Challenges

Medical services are provided into controllable environments, such as hospitals, by highly specialized medical professionals, with years of education and formal training, using very sophisticated biomedical equipment to perform specific treatments and cures, such as in acute care wards. The specialized facilities, the trained and experienced personnel, and the advanced biomedical equipment are scarce resources in Healthcare planning [1.43] because they increase drastically the associated costs with reduced or conditional availability in accessibility and without sometimes achieving the desired level of services. Hence, the motivation for the present research work steams from the fact that there are limiting barriers related with the healthcare resources along with the availability and accessibility of medical services.

The challenging issues related to home healthcare fall into two classes: first, the issues related with the functionality of the home healthcare model [1.44] and the ways patients can be treated at home, and second, the issues related with the development of a proper ubiquitous computing framework to facilitate the provision of healthcare services at home [1.45]. Hence, there is missing a unified and integrated Home Healthcare model to address (i) all kinds of diseases for (ii) all ages including infants and the elderly. Moreover, there is a lack of an effective and formally defined infrastructure and tools for the description of the Home Healthcare model’s functionality and its collaboration with the rest of the Health System’s components to provide a sustainable perspective. An adequate infrastructure with tools is required to rigorously and formally define Ubiquitous Computing models for Home Healthcare describing software infrastructures, operations, functionalities, and behaviors.

1.7 Objective Goals and Scope

The scientific community strives to develop feasible approaches to provide healthcare services in regular environments and to prevent sickness. Employing technological achievements and taking advantage of the pervasive software’s capabilities, home can be placed in the flow of medical information of the physical, the informational, and the cognitive domains [1.46]. The performed research efforts almost exhausted the physical and the information domains while the cognitive domain still remains to be explored. In other words, home healthcare needs an ubiquitous computing system providing cognition permitting the interaction with abstractions such as concepts and behavior [1.47]. Therefore, the objectives of the current research is the modeling of an infrastructure combining and integrating the medical cognitive domain with the physical and the informational for the ubiquitous participation of all required parties to provide healthcare at home.

The required hardware interfaces among the wirelessly networked devices through internet and the user interfaces of the remotely located collaborating parties for the development of the Home UbiHealth model are beyond the scope of this work. Within the scope of this work is the rigorous and formal description of the required software infrastructure to support the Home UbiHealth model using Denotational Mathematics.

1.8 Contributions

The performed research focuses on the modeling and the description of proper infrastructure that provides those tools to the scientific community to formally describe the provision of healthcare services at home. The contributions refer to:

1) An infrastructure that receives physical and semantic signals from the ubiquitous computing environment, interactions with the system inside and outside of home, and transforms them into interrelated and interacting medical concepts that eventually guide the supporting devices and the involved medical professionals at home. The formal description of the infrastructure provided using the expressiveness of Denotational Mathematics.
2) The associated administering tools of the infrastructure to control and manipulate medical concepts of varying level of detail.
3) The means to describe and control the software dynamic behavior with the principles of mobility, autonomy, migration, and polymorphism.

1.9 Thesis Outline

The rest of the research work is divided into ten more chapters the following parts:

1) Chapter 2: provides an overview of the environment of home healthcare and examines the conceptual background of a model of home care applications along with the intrinsic principles, ideas, and approaches.

2) Chapter 3: presents the dimensions of the formal framework of Denotational Mathematics required for the description of home healthcare models.
3) Chapter 4: refers to the required architectural components, the necessary tools, the presented capabilities to describe the static and dynamic behaviors, and the autonomous functionality of the home healthcare model. The model administers the involved medical knowledge to satisfy the personalized healthcare needs at home in a ubiquitous computing environment.
4) Chapter 5: examines the raised issues related with the management of the software applications operating in a ubiquitous computing environment. The operating software applications migrate from a device to another, adapt to the demands preserving with polymorphism their functionality, and adjust the supporting mechanisms to achieve personalized medical treatment through the administration of the involved medical knowledge.
5) Chapter 6: concludes the research work describing the obtained results from the application of a typical home ubihealth software system and discuss the opportunities for further research in the field of home healthcare supported by computing systems.

CHAPTER 2. BACKGROUND IN UBIHEALTH

The term *Home UbiHealth* [2.1] refers to the provision of healthcare services facilitated by ubiquitous computing means at the domestic environment. Home UbiHealth is challenged to provide individualized healthcare services [2.2], participate in formal Primary Healthcare and substitute parts of hospitalization [2.3]. UbiHealth is an active research area, which is expected to alter the attitude of medicine towards the provision of Primary Healthcare and hospitalization at home by integrating the physical, informational and cognitive domains of Information Systems [2.4].

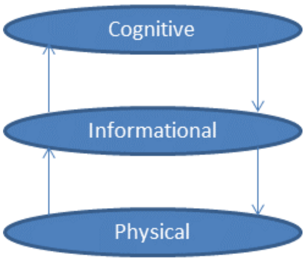


Figure-2.1 UbiHealth inside and outside of home.

2.1 The vision

Sensors, actuators, and computing devices constitute the physical domain. Practically, they are met everywhere, being "*woven into fabric*" as mentioned by Mark Weiser [2.5]. The processed data of the physical domain feeds the informational domain. Flowing information of the latter constitutes the input information the cognitive domain which is consisted of concepts as illustrated in Figure-2.1.

The idea behind ubiquitous computing is the eventual disappearing of computers, which stay in the background, supporting the user who concentrates on the execution of the performed task instead of the manipulation of a device in order to perform the task. Sensors, actuators, and computing devices are interconnected into overlapping but transparently cooperating ad-hoc networks [2.6] as illustrated in Figure-2.2. They cover the entire span from the patient's body (BAN) to personal (PAN), local (LAN), wide areas (WAN), and metropolitan areas (MAN) networks. Hence, the envisioned system's components must include networked devices, software, and the patient too.

The Home UbiHealth model supports the personalized patient's healthcare needs. Simultaneously, the model supports the connected locally or remotely located treating medical professionals who interact with the backing computing system, at the required each time level of detail in a ubiquitous and natural way. The Home UbiHealth model can embed approaches commonly met in the Internet of Things (IoT) [2.7] and the Cloud Computing [2.8] paradigm. Thus, the patient at home is set at the center of a virtual triangle with three domain poles of healthcare services, i.e., the physical, the informational, and the cognitive one. The Home UbiHealth model's output is the patient's health status [2.9] considering a compound structure of the medical, operational, functional, personal, financial, timing, security, and safety prerequisites. Therefore, the aim of the Home UbiHealth vision is the feasible quality

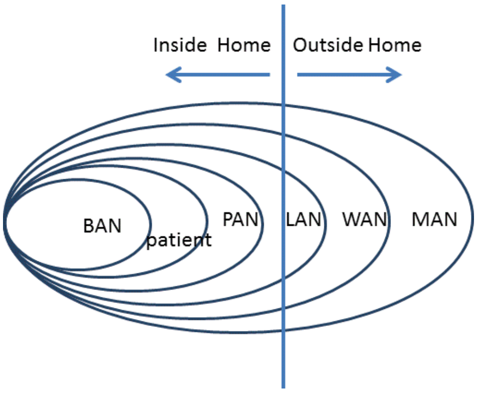


Figure-2.2 UbiHealth inside and outside of home

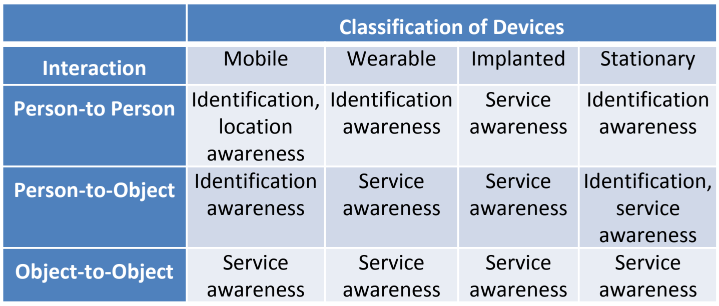
level of personalized care treating an individual in the domestic environment through a formally described system with the associated operational and functional tools.

2.2 Relevant Terminology

Some of the terms, directly or indirectly, related with the UbiHealth model are:

- **Ubiquitous computing:** Mark Weiser in the seminal article published in Scientific America in 1991, states: "*the most profound technologies are those that disappear*. *They weave themselves into the fabric of everyday life until they are indistinguishable from it*" [2.2]. The computing devices are less evidently conspicuous without coercing for attention and interaction. In that sense, ubiquitous computing becomes the underlying supporting principle for the ubiquitous provision of healthcare services at home.
- **Virtual Reality:** This is the opposite of ubiquitous computing. According to this paradigm the computing devices are disappearing but the users' sensing and affecting functionality is captured by the hosting computers [2.4] with the users entering into the computers' world. On the contrary, according to the ubiquitous computing paradigm, the computers enter in the users' world invisibly and unobtrusively [2.3].
- **Context:** According to Dey the context consists of all information that identifies and describes the state of the developed situations of the participating entities, which can be of any type such as artificial entities, software applications, physical components, and persons [2.1].

- **Types of Context:** These are classifications according to the acquiring source of data [2.5], the kind of exchanged data [2.6], the representation of the model [2.7], the nature and the kind of contextual synthesis [2.8], the types of the involved magnitudes [2.9], the observed state [2.10], the type of structures [2.10], and the kind of involved components [2.10].
- **Design types of context:** In the relevant literature, the design methods fall: (a) key-valued schemes dedicating variables for each contextual parameter, (b) markup schemes employing hierarchical data-structures, (c) graphical schemes facilitated by graphs, (d) object oriented schemes taking advantage of object properties, (e) logic based schemes utilizing logic ordered systems and inference rules, and (f) ontology based schemes relying on knowledge representation structures and specific taxonomies [2.11].
- **Context-awareness:** This is the property of the software applications to react autonomously supporting the user's tasks by adjusting their behavior to satisfy the occurring needs [2.1]. Awareness classifies context with active or passive behaviors [2.12] and it must be facilitated by an adequate infrastructure [2.13]. The property requires the consideration of every static or dynamic aspect of the surroundings, including as well as the supported user [2.14].
- **Activity:** It considers the situations of occurring events and the interactions among the involved persons and the participating objects or artifacts of hardware and software nature. The formal description of such interactions, as noted in the research literature, initiated by Leontev [2.15] and evolved further by Engestrom [2.16] delivering a hierarchical structure escorted by the solid theoretical basis of Activity Theory. For the activity manipulation in contextual analysis, an activity framework is required along with the proper methods and software tools. Hence, the integration of ubiquitous computing into the activities is achieved, while revealing and managing the collaboration among coexisting contexts [2.17]. The supporting activity framework, using tags, aims at the recognition, identification, and prediction of the performed activities with proper activity inferences [2.18].
- **Inference:** It relies on recognizing the performed activity which is basically carried out by analyzing the performed tasks as well as extracting parameters and characteristics. This is attained through models of patterns, models of probability, or temporal evidence theory. [2.19]. The employed models for activity recognition and inference require prior knowledge (Bayesian models [2.21]),



	Classification of Devices			
Interaction	Mobile	Wearable	Implanted	Stationary
Person-to Person	Identification, location awareness	Identification awareness	Service awareness	Identification awareness
Person-to-Object	Identification awareness	Service awareness	Service awareness	Identification, service awareness
Object-to-Object	Service awareness	Service awareness	Service awareness	Service awareness

Table-2.1 Classes of used devices in the functional interactions of persons and objects.

training (Markov models [2.20]), and memory storage (Evidence models, [2.21] - [2.25]).

- **Home healthcare:** It provides holistic solutions covering the individual's needs treated at home. It is adapted each time to the personal needs and preferences supporting a healthy daily living culturally accepted with security, social inclusion, and quality of life. Healing at a personal hospice, Senator Edward Kennedy, 1978, stated that "*Hospice is many things. Hospice is home care with inpatient back-up facilities. Hospice is pain control. Hospice is skilled nursing. Hospice is a doctor and a clergyman coming to your home…But most of all, hospice is the humanization of our health care system.*". In other words, home healthcare develops a model that can integrate the medical, psychological, social, and cultural needs of the individuals efficiently, i.e., with controlled financial costs, acceptable level of quality of care, and more effective use of technology. In 1967 the American Academy of Pediatrics adopted the term "*medical home*" [2.26] to express the embedded characteristics of an involved personal physician who directs a team of medical practice oriented to face the whole person's problems and coordinates the medical and nursing services across all levels of the Healthcare System achieving an improved quality and security relying on evidence-based medicine. Home healthcare is considered to be an extended model of *medical home* [2.27] with additional organizational attributes. This model relies on the efficient transfer of information which is integrated in the Healthcare Systems' services. Such services are provided at home with continuous involvement of medical professionals along with all medical information for the provision of adequate medical treatments [2-28].

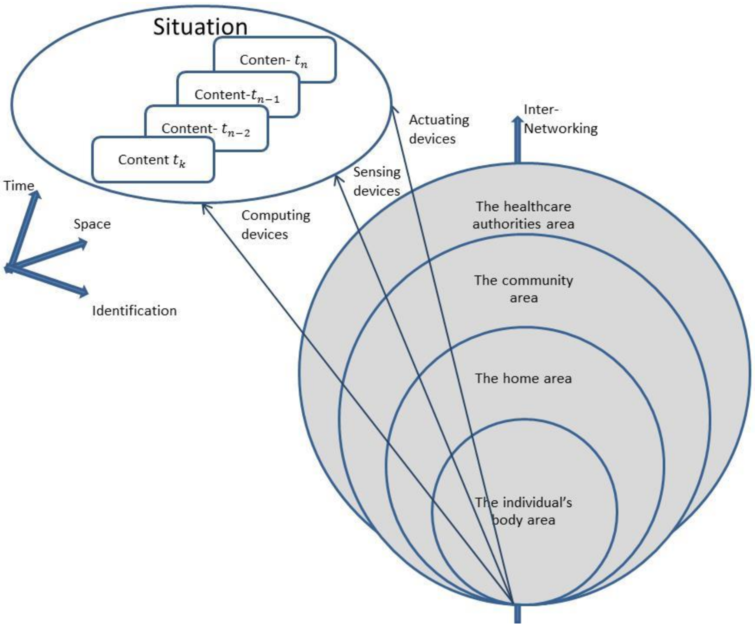


Figure-2.3. A representative model's structural abstraction along some dimensions.

- **<u>Home UbiHealth:</u>** It is about a technological paradigm for the provision of healthcare services in the ubiquitous computing environment at home referred as "*Home UbiHealth*". The major objective of this paradigm is the introduction of computers in the users' real world, operating in an unobtrusive manner and assisting to achieve the proper healthcare provision in the home environment while supporting the treated patients' autonomy [2.29] at the desired domestic premises. The home UbiHealth is approached from the infrastructure and the services side as follows:

  - **<u>Facilitating Infrastructure (Hardware):</u>** It provides the means to implement the home UbiHealth vision by scaling down the size of the required sensors, the necessary computing devices, and the associated medical devices at the scale of nano-meters. These are called smart dust motes due to their characteristic small dimensions. The networking multi-hop ad-hoc capabilities create dense networks of smart devices, along with micro-electro-mechanical (MEM) devices of proportional size, with each node to retransmit – repeat the carried messages facilitating the continuous enriched flows of information with the conventional ICT systems [2.30]. The employed devices can be uniquely identified and tagged, implementing additional paradigms such as "*things that think*" (TTT) [2.31] and internet of things (IoT) [2.32] in order to provide the necessary interoperability for the development of medical, clinical, and business supporting workflows [2.33]. The intrinsic intelligence of the employed devices allows the adaptation of their operations to the personal preference or the personal healthcare needs of the patient [2.34]. The coexistence of many sources of information introduces ambiguity in the cases of contradictory or disagreeing evidential measurements, thereby revealing the need for data and information fusion [2.35] to support decision making. The performed operations among participating persons and coexisting devices are summarized in Table-2.1 regardless of the method (e.g., software as a service (SAAS), infrastructure as a service (IAAS) or platform as a service (PAAS) [2.37]) [2.36]. The infrastructure transparently provides autonomous support anywhere and anytime in spite of the energy constraints [2.38].

- **Services (Software):** This includes the systemic view of health, which consists of distinct and discretely identified states and the relevant conditions [2.39, 2.40]. The provision of healthcare services can be distinguished into the provision of personalized healthcare services for the satisfaction of each individual's health needs and personal preferences [2.41], the capability to adapt in the holding conditions of the occurring situations [2.42, 2.43], and the category of the supported individual [2.44]. Such considerations provide the distinction of the offered services into three categories: (i) emergency, (ii) assist living, and (iii) improving quality of life. The categories of the offered services at home must correspond to the patient categories, which can be classified into the 3 classes given in Table-1.2. The provided healthcare services at home must satisfy the prerequisites of privacy and security [2.45] to achieve the proper performance of the involved roles [2.46].

- **Personal well-being**: It refers to evidence-based medicine and evidence-based management for the support of medical professionals to reduce errors, and augmenting quality. Also, the clinical practice instrumented with IT means strives for reliable evidences that equip medical professionals with security [2.47].

2.3 Model Abstraction

Home UbiHealth can be viewed to embrace the following two infrastructures:

A. **Hardware infrastructure:** With respect to mobility, hardware infrastructure can be distinguished as: (a) mobile (either carried, i.e. in-vitro, or implanted, i.e.in-vivo), and (b) stationary. Moreover, with respect to functionality it can be classified into (i) sensing, (ii) actuating, (iii) computing, and (iv) hybrid. The infrastructure provides ad-hoc networking to the participating devices achieving the desired home UbiHealth aims. In Figure-2.3, a representative high-level of abstraction is outlined.

The constituents of the model's abstraction, as shown in Figure-2.3, include:

- **Sensors:** They measure pre-determined physical magnitudes
- **Processing unit:** It is a layered structure that provides the following services:
    - ► Input processing (pre-processing): receives, stores, and excludes uncertainty.
    - ► Processing: classifies and transforms raw data into information, including:
        - ■ Tagging: provide identification stamps on the received data.
        - ■ Fussing: dissolve ambiguities and decides on the dominating values.
        - ■ Storing: memory storage after classifying the data accordingly.
        - ■ Extracting: applying sets of rules to spawn artificial parameters.
    - ► Output processing (meta-processing): sets available the processed results.
- **Memory storage:** It provides the capability to store, retrieve, and re-store in:
    - ► Short-term memory: minimizing fetching overhead.
    - ► Medium-term memory: the storage for the less frequently used data.
    - ► Long-term memory: historic data serving the retrieval procedures.
- **Actuators:** They provide the means to intervene and affect the individual's behavior:
    - ► Logic: computing decisions, issuing signals, or activating software.
    - ► Micro-Electro-Mechanical devices (MEMs): influencing situations.
- **Wireless communication:** among the dispersed and interoperate devices.

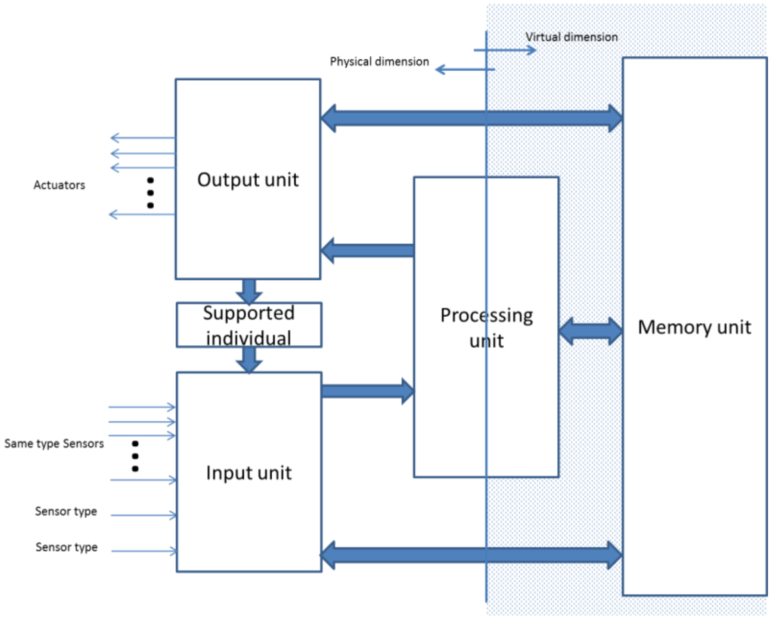


Figure-2.4. Representative block diagram.

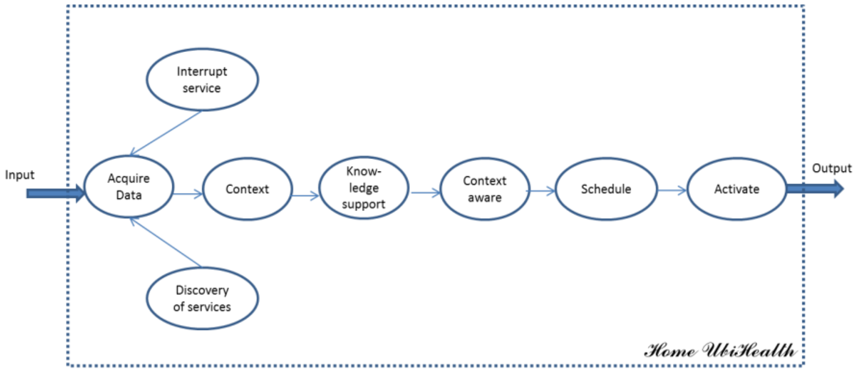


Figure-2.5 The major functionalities of an abstraction for Home UbiHealth.

The building blocks of Figure-2.4 show the input measurements, *S*, of physical magnitudes from single or multi-sensors,

$$S = \{s_1, s_2, s_3, \dots, s_n\} = \{(s_1, s_2, \dots, s_m), \dots, (s_k, s_{k+1}, \dots, s_{k+o}), \dots, s_n\} = \bigcup_{k=1}^{n} S_k \quad \text{(Eq-2.1)}$$

The actuators fall into two major categories, the physical actuators, *P*, (e.g., MEMs) and software actuators, *L*, (e.g., activating the execution of a software application),

$$A = P \; U \; L = \{p_1, p_2, \dots, p_k\} \; U \; \{l_1, l_2, \dots, l_m\} = (\bigcup_{i=1}^{k} P_i) \; \text{U} \; (\bigcup_{j=1}^{m} L_j) \quad \text{(Eq-2.2)}$$

The processing unit of the block diagram is represented by the composition of the individual functions that correspond to the pre-processing and the processing stages.

$$f{:}\, A \to R \; and \; x \to f(x) \qquad \text{(Eq. 2.3)}$$

$$g{:}\, B \to S \; and \; x \to g(x) \qquad \text{(Eq. 2.4)}$$

$$(gof)(x) \; = \; g\big(f(x)\big) \qquad \text{(Eq. 2.5)}$$

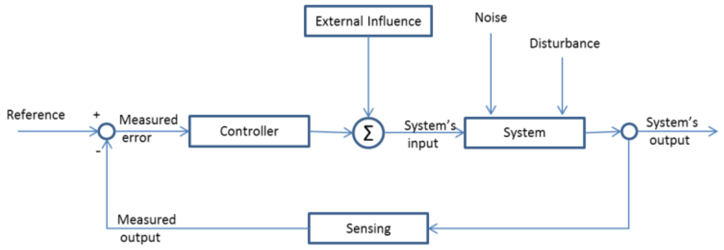


Figure-2.6. First-order, closed loop control for the home UbiHealth system.

The distinction of memory into short-, middle-, and long-term storage space is represented by multidimensional matrices which constitute the supporting memory to the formed system

$$\overline{M(\iota_1, \dots, \iota_n)} = \overline{M_{sh}(\iota_1, \dots, \iota_n)} + \overline{M_{m\iota d}(\iota_1, \dots, \iota_n)} + \overline{M_{long}(\iota_1, \dots, \iota_n)} \quad (2.6)$$

The home UbiHealth system evolves with transitions through discrete states corresponding to the dynamic behavior of the individual’s health as shown in Figure-2.5. Thus, the interaction of the system with the environment is monitored and control methods apply to determine the states of equilibrium which correspond to the individual’s health statuses. The interaction involves secure open connectivity, failures handling, transparent concurrency, and consistent operation within the boundaries considered by the medical doctors for the individual’s health.

B. **Software infrastructure:** It must perform the functionality presented in Figure-2.5, resolving issues related to naming, communication, synchronization, and scalability. It must be characterized by the principles of separation of concerns for the designer, adjusted granularity in the infrastructure’s modularity for making references to modules and components, conceptual abstraction for applying strategy and policies, and anticipation of change for proper adaptation.

The administration of the system requires continuously updated feedback as illustrated in Figure-2.6, forming a first-order, closed-loop applied on the infrastructure monitoring and controlling its functionality. The individual’s health status is supported by the system and, at the same time, the individual observes and affects its operation. In other words, the individual is observing and simultaneously observed by the system. This implies that the individual’s reference is inside and outside of the system, requiring the controlling and the controlled factors to match at all times for an acceptable operation.

The functionality of the supporting infrastructure of Figure-2.5, must include:

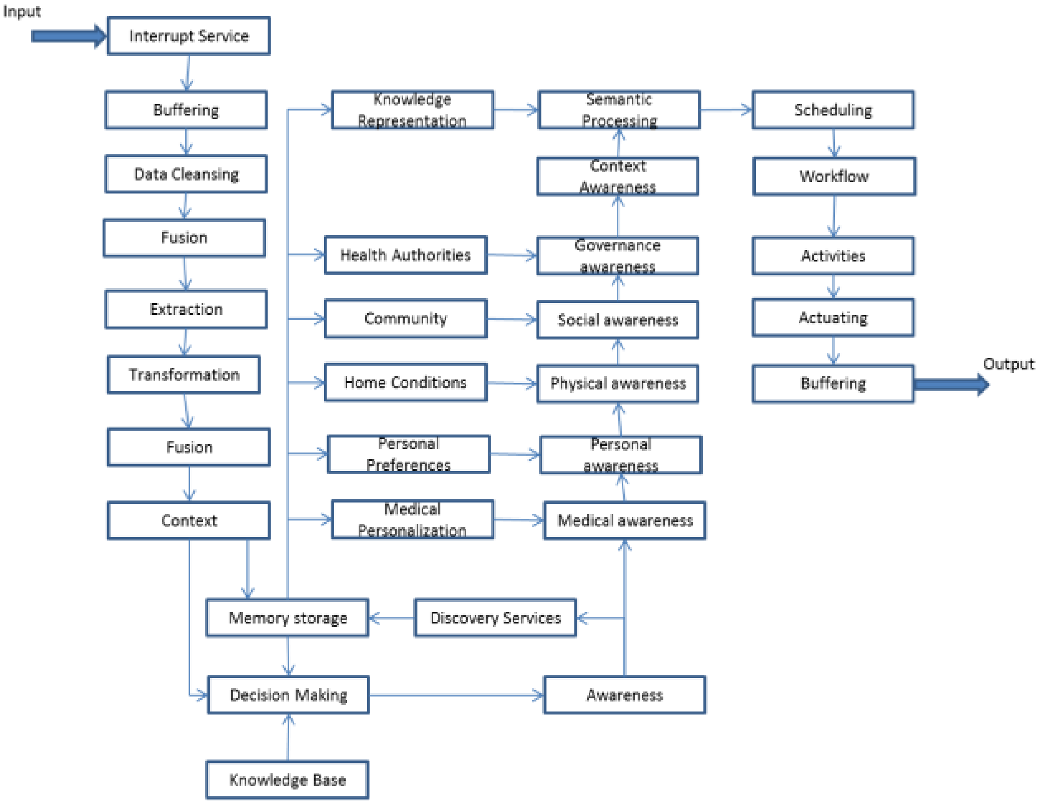


Figure-2.7 The interaction of components in the home UbiHealth system.

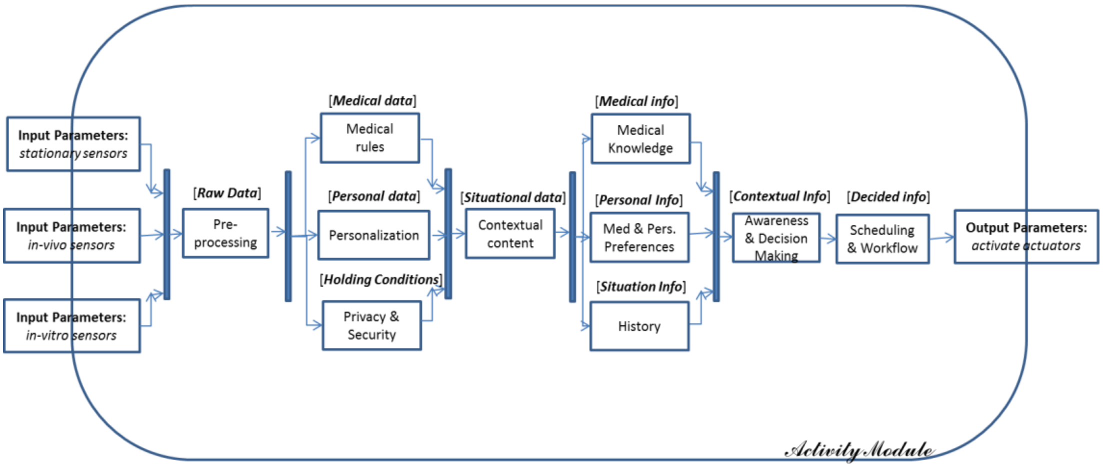
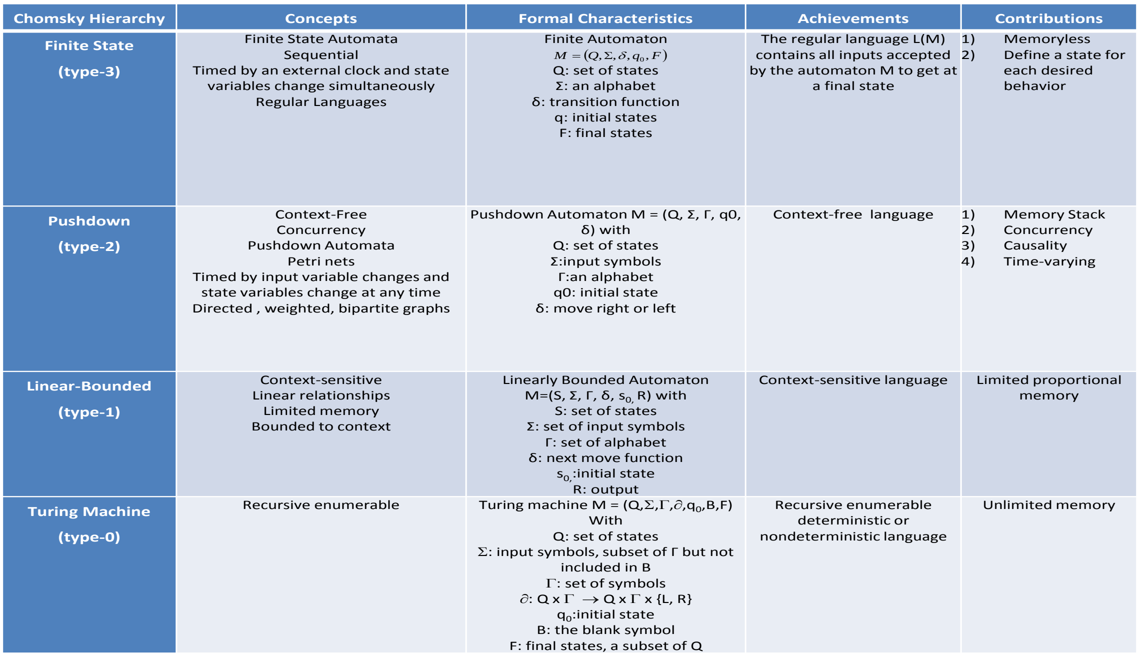


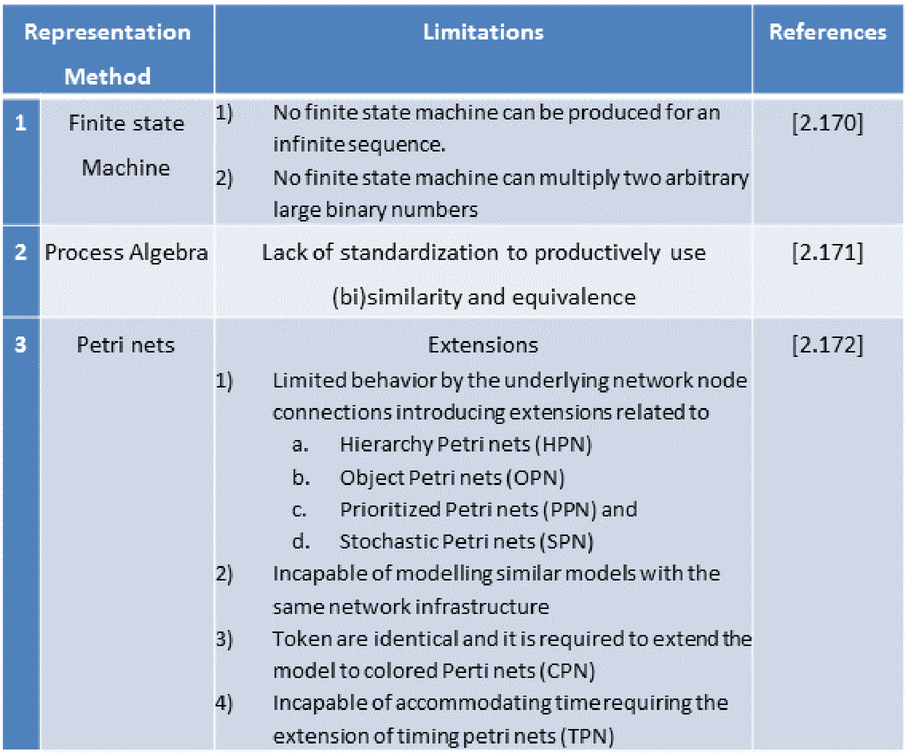
Figure-2.8 Abstraction of a UML activity module of home UbiHealth system.

- Acquisition: buffering, data cleansing, fusion, extraction, and transformation.
- Context: contextual content, memory storage, decision making, knowledge base.
- Knowledge: representation, policies, preferences, medical personalization.
- Context aware: medical, personal, physical, social, governance.
- Schedule: scheduling tasks, operations, procedures, processes and workflows.
- Activation: recognition and actuation.
- Services: interrupt handling and discovery services for devices and components



Chomsky Hierarchy	Concepts	Formal Characteristics	Achievements	Contributions
Finite State **(type-3)**	Finite State Automata Sequential Timed by an external clock and state variables change simultaneously Regular Languages	Finite Automaton $M = (Q, \Sigma, \delta, q_0, F)$ Q: set of states Σ: an alphabet δ: transition function q: initial states F: final states	The regular language L(M) contains all inputs accepted by the automaton M to get at a final state	1) Memoryless 2) Define a state for each desired behavior
Pushdown **(type-2)**	Context-Free Concurrency Pushdown Automata Petri nets Timed by input variable changes and state variables change at any time Directed , weighted, bipartite graphs	Pushdown Automaton M = (Q, Σ, Γ, q0, δ) with Q: set of states Σ:input symbols Γ:an alphabet q0: initial state δ: move right or left	Context-free language	1) Memory Stack 2) Concurrency 3) Causality 4) Time-varying
Linear-Bounded **(type-1)**	Context-sensitive Linear relationships Limited memory Bounded to context	Linearly Bounded Automaton M=(S, Σ, Γ, δ, $s_{0,}$ R) with S: set of states Σ: set of input symbols Γ: set of alphabet δ: next move function $s_{0,}$:initial state R: output	Context-sensitive language	Limited proportional memory
Turing Machine **(type-0)**	Recursive enumerable	Turing machine M = ($Q,\Sigma,\Gamma,\partial,q_0$,B,F) With Q: set of states Σ: input symbols, subset of Γ but not included in B Γ: set of symbols ∂: Q x Γ → Q x Γ x {L, R} q_0:initial state B: the blank symbol F: final states, a subset of Q	Recursive enumerable deterministic or nondeterministic language	Unlimited memory

Table-2.2 Major formal description approaches for ubiquitous computing.



	Representation Method	Limitations	References
1	Finite state Machine	1) No finite state machine can be produced for an infinite sequence. 2) No finite state machine can multiply two arbitrary large binary numbers	[2.170]
2	Process Algebra	Lack of standardization to productively use (bi)similarity and equivalence	[2.171]
3	Petri nets	Extensions 1) Limited behavior by the underlying network node connections introducing extensions related to a. Hierarchy Petri nets (HPN) b. Object Petri nets (OPN) c. Prioritized Petri nets (PPN) and d. Stochastic Petri nets (SPN) 2) Incapable of modelling similar models with the same network infrastructure 3) Token are identical and it is required to extend the model to colored Perti nets (CPN) 4) Incapable of accommodating time requiring the extension of timing petri nets (TPN)	[2.172]

Table-2.3 Limitations of the relevant approaches.

The distributed nature of the infrastructure is presented in Figure-2.7. The system's activity can be presented with a Unified Modeling Language (UML) diagram independently from coding and platform. Thus, functional medical requirements can be included in a model-driven architecture (provided by OMG at www.omg.org), given representations of the underlying medical knowledge [2.48] with medical ontologies [2.49]. Activity diagrams as in Figure-2.8 are incapable to present the system's distributed nature due to required knowledge support to activate and guide the dispersed devices [2.49].

2.4 Related Design Approaches

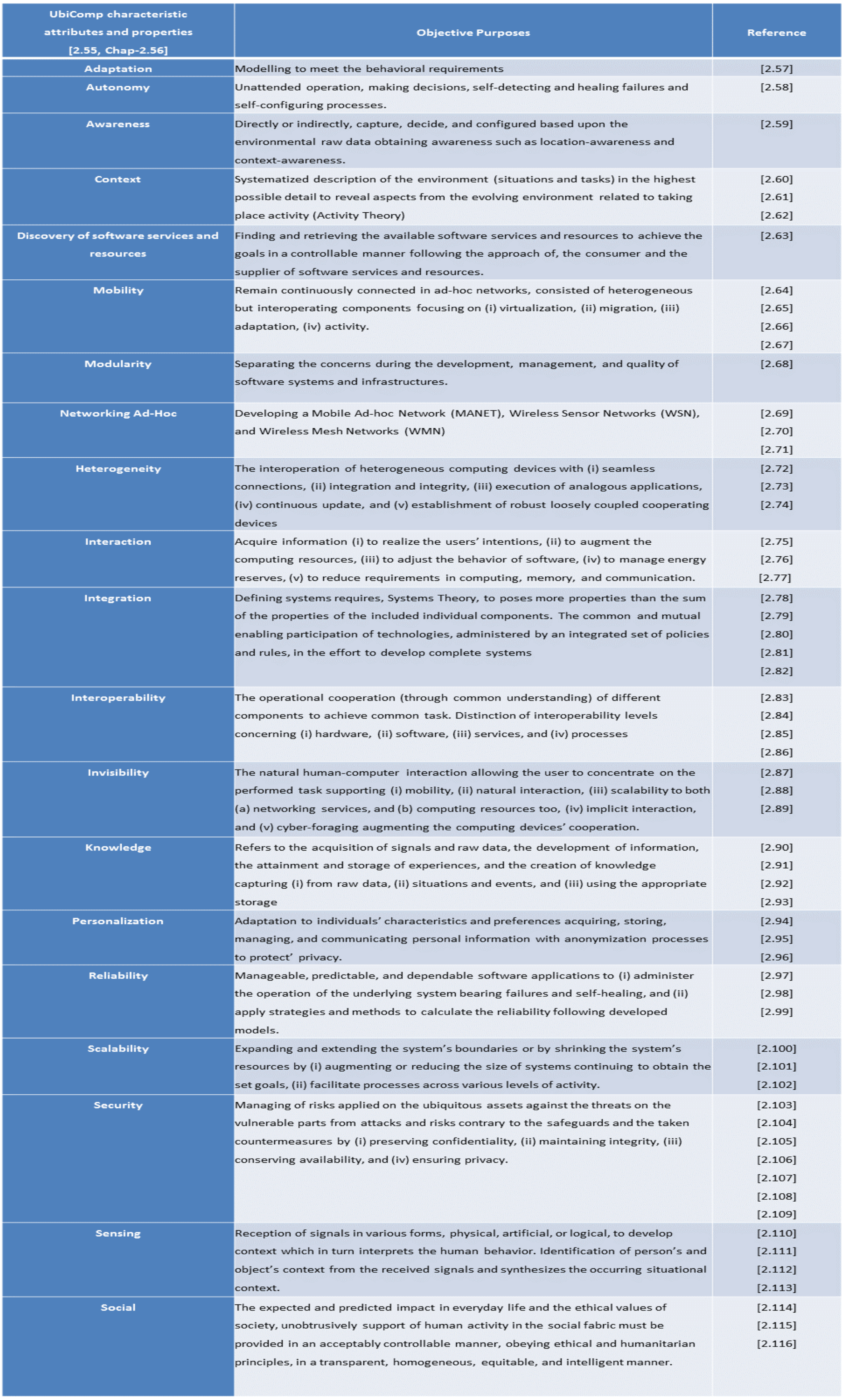
The Chomsky Hierarchy [2.50] provides a classification of languages with the corresponding grammar and its supporting automaton as reported in Table-2.2. In this framework, language is analogous to computing operational behavior, grammar corresponds to applied rules, and the automata are analogous to the supporting hardware infrastructures.

Formal modelling approaches as summarized in Table-2.2 involve either the use of a language based on formally defined grammar and appropriate production rules (type-0 through type-3), the use of formally defined algebraic frameworks consisted of syntactic rules for handling processes (type-1), or the use of expressive and formally defined graphical media representing processes evolution describing states and transition states (type-0). There are three modelling directions [2.51],

- Finite State Machines: (type-3 and type-2) adopting the Unified Modeling Language (UML) standard.
- Process Algebra: (type-1) syntactic rules, semantic mapping, and processes equivalence.
- Petri Nets: (type-0) a diagrammatic tool for modeling concurrency and synchronization.

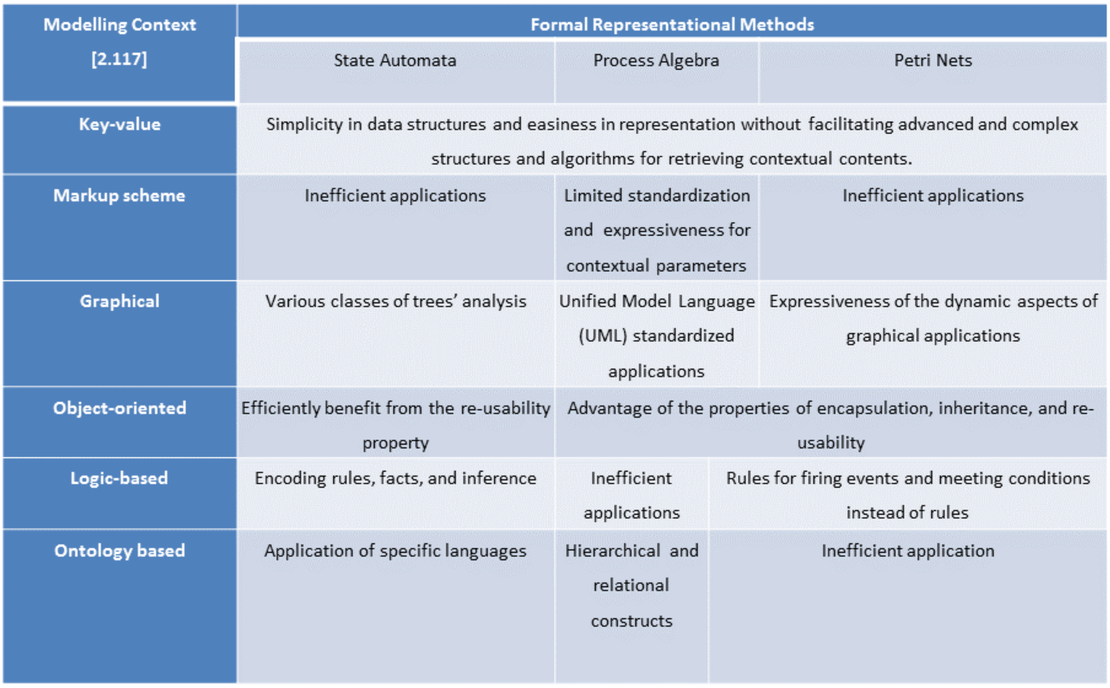
The limitations of the described approaches are summarized in Table-2.3.

Table-2.4 provides characteristics of the UbiComp paradigm. The related literature classifies the context models into categories according to the employed data structures with the corresponding noticed limitations as it is shown in Table-2.5. From the comparison of the representation methods against the context models, it is shown that formal descriptions of software applications are provided only in certain cases, underlining the lack of universal formal description to represent contextual contents.



UbiComp characteristic attributes and properties [2.55, Chap-2.56]	Objective Purposes	Reference
Adaptation	Modelling to meet the behavioral requirements	[2.57]
Autonomy	Unattended operation, making decisions, self-detecting and healing failures and self-configuring processes.	[2.58]
Awareness	Directly or indirectly, capture, decide, and configured based upon the environmental raw data obtaining awareness such as location-awareness and context-awareness.	[2.59]
Context	Systematized description of the environment (situations and tasks) in the highest possible detail to reveal aspects from the evolving environment related to taking place activity (Activity Theory)	[2.60] [2.61] [2.62]
Discovery of software services and resources	Finding and retrieving the available software services and resources to achieve the goals in a controllable manner following the approach of, the consumer and the supplier of software services and resources.	[2.63]
Mobility	Remain continuously connected in ad-hoc networks, consisted of heterogeneous but interoperating components focusing on (i) virtualization, (ii) migration, (iii) adaptation, (iv) activity.	[2.64] [2.65] [2.66] [2.67]
Modularity	Separating the concerns during the development, management, and quality of software systems and infrastructures.	[2.68]
Networking Ad-Hoc	Developing a Mobile Ad-hoc Network (MANET), Wireless Sensor Networks (WSN), and Wireless Mesh Networks (WMN)	[2.69] [2.70] [2.71]
Heterogeneity	The interoperation of heterogeneous computing devices with (i) seamless connections, (ii) integration and integrity, (iii) execution of analogous applications, (iv) continuous update, and (v) establishment of robust loosely coupled cooperating devices	[2.72] [2.73] [2.74]
Interaction	Acquire information (i) to realize the users' intentions, (ii) to augment the computing resources, (iii) to adjust the behavior of software, (iv) to manage energy reserves, (v) to reduce requirements in computing, memory, and communication.	[2.75] [2.76] [2.77]
Integration	Defining systems requires, Systems Theory, to poses more properties than the sum of the properties of the included individual components. The common and mutual enabling participation of technologies, administered by an integrated set of policies and rules, in the effort to develop complete systems	[2.78] [2.79] [2.80] [2.81] [2.82]
Interoperability	The operational cooperation (through common understanding) of different components to achieve common task. Distinction of interoperability levels concerning (i) hardware, (ii) software, (iii) services, and (iv) processes	[2.83] [2.84] [2.85] [2.86]
Invisibility	The natural human-computer interaction allowing the user to concentrate on the performed task supporting (i) mobility, (ii) natural interaction, (iii) scalability to both (a) networking services, and (b) computing resources too, (iv) implicit interaction, and (v) cyber-foraging augmenting the computing devices' cooperation.	[2.87] [2.88] [2.89]
Knowledge	Refers to the acquisition of signals and raw data, the development of information, the attainment and storage of experiences, and the creation of knowledge capturing (i) from raw data, (ii) situations and events, and (iii) using the appropriate storage	[2.90] [2.91] [2.92] [2.93]
Personalization	Adaptation to individuals' characteristics and preferences acquiring, storing, managing, and communicating personal information with anonymization processes to protect' privacy.	[2.94] [2.95] [2.96]
Reliability	Manageable, predictable, and dependable software applications to (i) administer the operation of the underlying system bearing failures and self-healing, and (ii) apply strategies and methods to calculate the reliability following developed models.	[2.97] [2.98] [2.99]
Scalability	Expanding and extending the system's boundaries or by shrinking the system's resources by (i) augmenting or reducing the size of systems continuing to obtain the set goals, (ii) facilitate processes across various levels of activity.	[2.100] [2.101] [2.102]
Security	Managing of risks applied on the ubiquitous assets against the threats on the vulnerable parts from attacks and risks contrary to the safeguards and the taken countermeasures by (i) preserving confidentiality, (ii) maintaining integrity, (iii) conserving availability, and (iv) ensuring privacy.	[2.103] [2.104] [2.105] [2.106] [2.107] [2.108] [2.109]
Sensing	Reception of signals in various forms, physical, artificial, or logical, to develop context which in turn interprets the human behavior. Identification of person's and object's context from the received signals and synthesizes the occurring situational context.	[2.110] [2.111] [2.112] [2.113]
Social	The expected and predicted impact in everyday life and the ethical values of society, unobtrusively support of human activity in the social fabric must be provided in an acceptably controllable manner, obeying ethical and humanitarian principles, in a transparent, homogeneous, equitable, and intelligent manner.	[2.114] [2.115] [2.116]

Table-2.4 Ubiquitous computing characteristics.



Modelling Context [2.117]	Formal Representational Methods		
	State Automata	Process Algebra	Petri Nets
Key-value	Simplicity in data structures and easiness in representation without facilitating advanced and complex structures and algorithms for retrieving contextual contents.		
Markup scheme	Inefficient applications	Limited standardization and expressiveness for contextual parameters	Inefficient applications
Graphical	Various classes of trees' analysis	Unified Model Language (UML) standardized applications	Expressiveness of the dynamic aspects of graphical applications
Object-oriented	Efficiently benefit from the re-usability property	Advantage of the properties of encapsulation, inheritance, and re-usability	
Logic-based	Encoding rules, facts, and inference	Inefficient applications	Rules for firing events and meeting conditions instead of rules
Ontology based	Application of specific languages	Hierarchical and relational constructs	Inefficient application

Table-2.5 Relating context models with formally describing methods.

2.5 Determinant Home UbiHealth Factors

The healthcare stereotypes of the western societies, as shown in Figure-2.9, are organized in, (i) the primary care, (ii) the secondary care, and (iii) the tertiary care. Primary Care is the first level of every Healthcare System and brings together the citizens with organized healthcare services where the applied policies enforce equity of access as well as face social and financial disparities. The sustainability of the Healthcare Systems is affected, as it follows, by the Social Security System, the Primary Care model, and the organizational model supporting the involved flow of information.

2.5.1 Types of Healthcare Systems

Every organized society forms and arranges the institutions, instruments, and tools to maintain the covered population healthy, treat the sick, and protect individuals and families from the financial burden of medical expenses. There are four basic models for Health Care and each country applies variations or mixtures depending upon the local cultural and financial conditions (www.who.int):

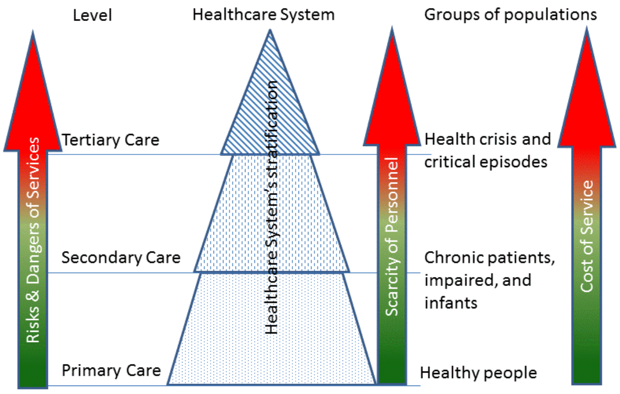

Level
Healthcare System
Groups of populations
Tertiary Care
Secondary Care
Primary Care
Risks & Dangers of Services
Healthcare System's stratification
Scarcity of Personnel
Cost of Service
Health crisis and critical episodes
Chronic patients, impaired, and infants
Healthy people

Figure-2.9 Levels of Care and the corresponding supported populations.

1. Beveridge Model: It was introduced by the Beveridge Report in 1942.
2. Bismarck Model: It was named after the Prussian chancellor Otto van Bismarck.
3. National Health Insurance Model: It is a combination of the previous two models.
4. Out-of-Pocket Model: It is a model met at less organized countries and it refers to the direct financial contribution of the population for services.
5. Hybrid Model: characteristics and variations from the above known models.

This classification provides significant characteristics such as the single payer principle that improves control on services and costs, the establishment of Sickness Funds insuring the population's health, the Social Insurance institution considering the healthcare needs of the society, the existence of National Health Insurance providing a holistic approach for all aspects of population's healthcare, and the Out-of-Pocket paradigm that refers to a market driven healthcare system.

2.5.2 The Patient-Centered Medical Home (PCMH)

The term *Medical Home* was introduced by the American Academy of Pediatrics (AAP) in 1967 (www.aap.org) describing a methodological follow-up process applied by the American pediatricians. The *Medical Home* concept evolved, in year 2000, and standardized to include the following seven aspects-characteristics:

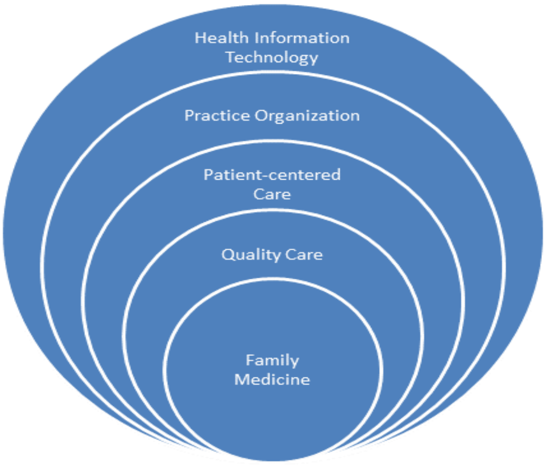


Figure-2.10. The Patent Centered Medical Home (PCMH) structure.

1. Accessibility: The procedures that update the contents of the medical records.
2. Continuity: The procedures that ensure the continuous updating.
3. Comprehensiveness: The reservation of every aspect of the healthcare status.
4. Family-centered: The medical records store data relevant to the individual’s family.
5. Coordination: The medical decisions rely on the maintained medical records’ data.
6. Compassion: The socially related data allow the support of the impaired individual.
7. Culture: The stored medical information includes culturally relevant data associated with the provided care.

In 2007 *Medical Home* was adopted by the American Academy of Family Physicians (AAFP, www.aafp.ord), the American College of Physicians (ACP, www.acponline.org), and other academies and associations. The evolving concept includes a dedicated physician who synchronizes specialists with holistic oriented health actions, providing coordination with quality and safety while ensuring accessibility and support by reimbursement services. Figure-2.10 depicts the structure of the Patent Centered Medical Home (PCMH, www.pcmh.ahrq.gov) model. Information Thechnology supports the functionality of the PCMH model achieving reduction of mortality and morbidity, properly usage of medication, and controlling the expenditures with patients’ satisfaction and equity in the health care access.

2.5.3 The Smart Home model

The term *Smart Home* refers to intelligent environments that adopt technological means to support the inhabitants to carry out their tasks. The technology at home administers knowledge to carry out tasks while self-adapting and adjusted to the held situations considering the inhabitants' security and privacy. The natural interaction with computers requires the participation and processing of the developed context.

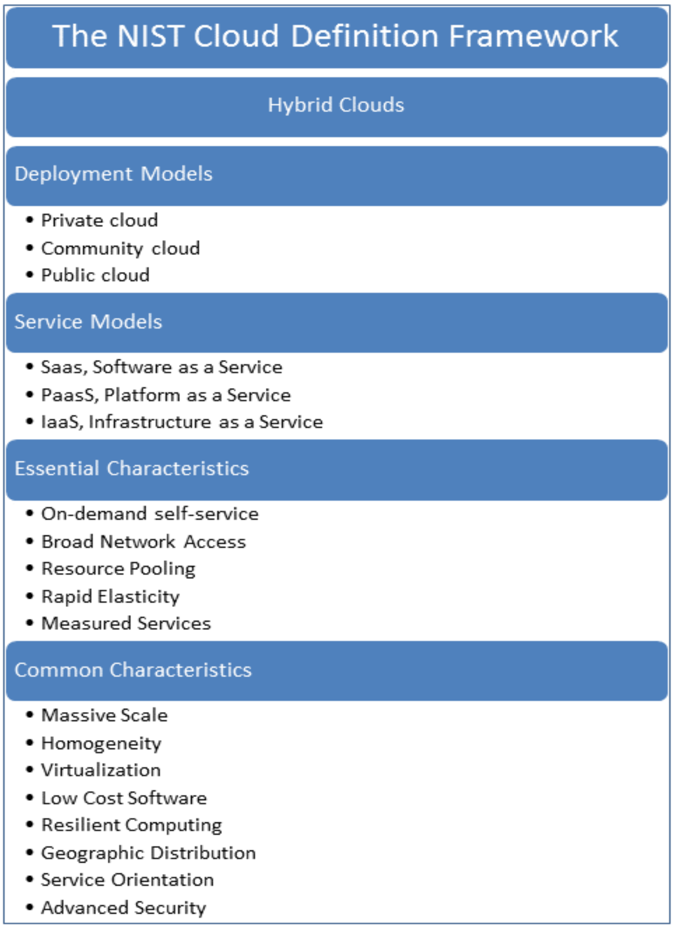
The academic community is concerned with the *Smart Home* model as in the cases,

1) The Aware Home: The research project developed at the Georgia Institute of Technology (www.cc.gatech.edu).
2) Intelligent Room: The project was performed at the Massachusetts Institute of Technology (www.ai.mit.edu).
3) Interactive Workspaces: The research work implemented at the Stanford University (www.iwork.stanford.edu).
4) Boulder Adaptive House: The project was performed by the University of Colorado (www.cs.colorado.edu).
5) MavHome Smart Home: The research test bed of the University of Texas at Arlington (www.uta.edu).
6) The Smart Home: The research work carried at the Texas Christian University (www.tcu.edu).

The industry occupied with the development of commercially available products, indicatively:

1) General Electric, the Smart Home (www.geindustrial.com).
2) Microsoft Co, the Microsoft Easy Living (www.microsoft.com).
3) Phillips Research, the Vision of the Future (www.design.phillips.com).
4) Verizon, the Connected Family (www.verizonwireless.com).

Hence, it is realized the feasibility of knowledge administration at home and its impact on the quality of the inhabitants' life and the sustainability of the model.



The NIST Cloud Definition Framework

Hybrid Clouds

Deployment Models

- Private cloud
- Community cloud
- Public cloud

Service Models

- Saas, Software as a Service
- PaasS, Platform as a Service
- IaaS, Infrastructure as a Service

Essential Characteristics

- On-demand self-service
- Broad Network Access
- Resource Pooling
- Rapid Elasticity
- Measured Services

Common Characteristics

- Massive Scale
- Homogeneity
- Virtualization
- Low Cost Software
- Resilient Computing
- Geographic Distribution
- Service Orientation
- Advanced Security

Figure-2.11. The NIST cloud definition framework.

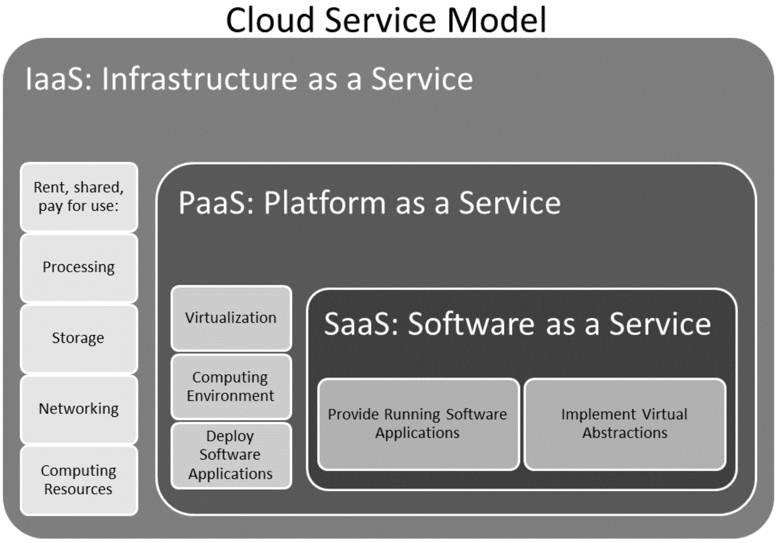

Cloud Service Model
IaaS: Infrastructure as a Service
Rent, shared, pay for use:
Processing
Storage
Networking
Computing Resources
PaaS: Platform as a Service
Virtualization
Computing Environment
Deploy Software Applications
SaaS: Software as a Service
Provide Running Software Applications
Implement Virtual Abstractions

Figure-2.12. Schematic relationships among the Cloud Service Models.

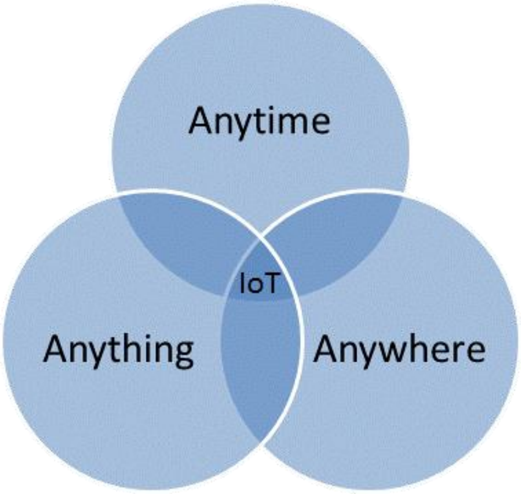


Figure-2.13. The intersection of time, space, and things.

2.5.4 Cloud Computing

Cloud Computing provides configurable anywhere/anytime services on-demand over the internet, hiding infrastructural complexity via interfaces that are (i) remotely hosted, (ii) ubiquitous, and (iii) commoditized services that can be measured and paid as utilities. Figure-2.11 presents the National Institute of Standards (www.nist.gov) definition.

Cloud computing achieves economies of scale because it lowers the initial investment allocating the services' costs on-demand. The providers of the services claim profits by the establishment of an ongoing stream of services the way gas or water utility is provided nowadays. However, availability, reliability, security and privacy issues are hidden from the end-user. Thus, it is transparent to the end-user how the involved data is administered on the user's behalf who is unable to participate in the resolution of failure problems of the provider of services.

Cloud computing paradigm as illustrated in Figure-2.12 presents advantages related to the stratification of services, thereby modulating the intensity of the offered services, allocating operational costs, and providing storage capacity, universal access, collaborations, and device independence. The disadvantages are related with the availability of internet connection, bandwidth, browser capabilities, poor interfaces, security, and standardization. Cloud computing provides services' massive consumption, facilitating the model of collaborative Cloud computing that approximates the limits of Grid computing.

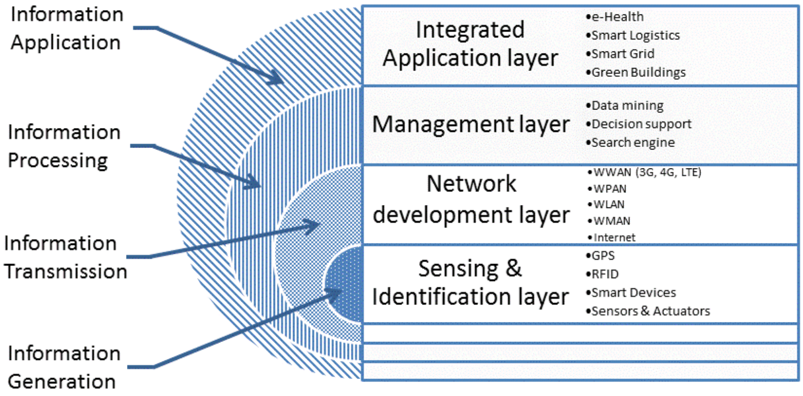


Figure-2.14. Stratification of the IoT model.

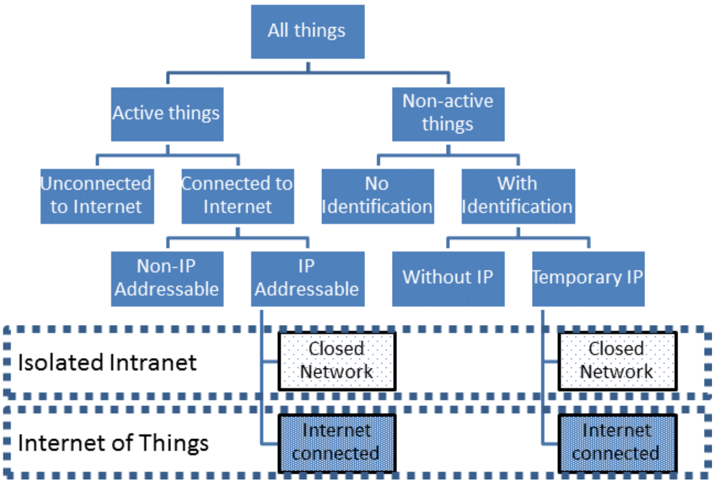


Figure-2.15 Classification of smart things.

2.5.5 Internet of Things (IoT) Paradigm

In smart spaces' models, objects are self-identified and expose characteristic attributes and properties through intelligent interfaces. The availability of wireless networks with short-range self-configured transceivers develops innovative interactions among things and people. Figure-2.13 presents the relationships of the physical magnitudes of time, space, and identity among the involved smart things.

The IoT paradigm involves devices connected to internet and participating in the carried procedures. Their dissemination raises concerns about privacy, security, governance, interoperability among wirelessly communicating heterogeneous self-configured devices, as well as identification and IP-addressing. The IoT model relies on a four layers structure as presented in Figure-2.14, synthesizing signals to complete information. The classification tree of *smart things* is depicted in Figure-2.15, leading to the Internet of Everything (IoE) model, which is an extension of the IoT.

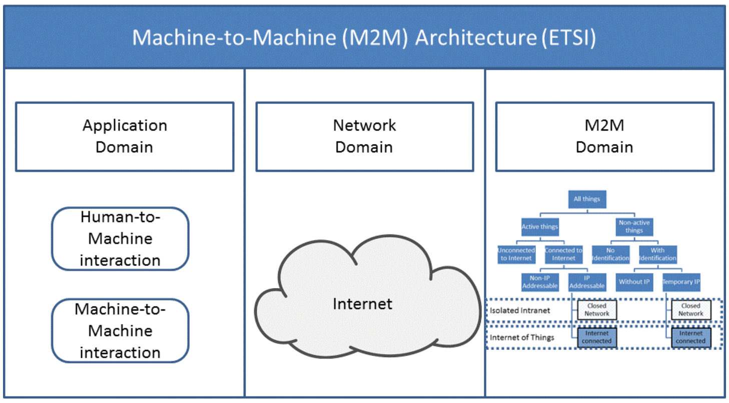


Figure-2.16. The Machine-to-Machine generic architecture according to ETSI.

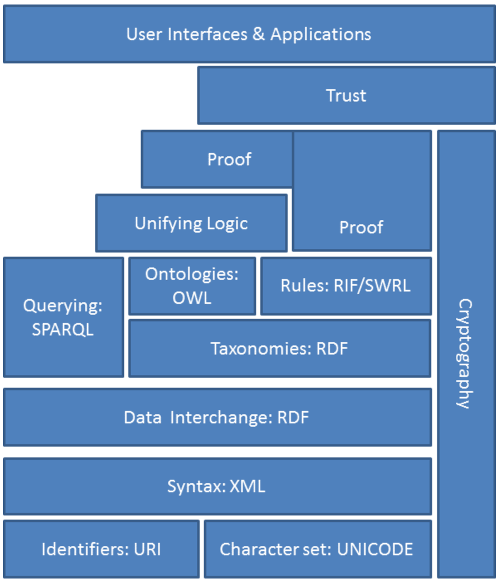


Figure-2.17. The generic architectural stack and standards turning data into knowledge.

The collaboration among smart things requires an architectural abstraction consisted of three layers as presented in Figure-2.16, which adheres to the machine–to-machine (M2M) generic principles suggested by the European Telecommunications Standards Institute (ETSI). The IoT model develops complete information and knowledge assisted by ontologies and logic. Figure-2.17 depicts a software stack integrating data into knowledge. The transformation process that turns raw data into usable knowledge contains at least three layers driven by software applications as can be seen in Figure-2.18.

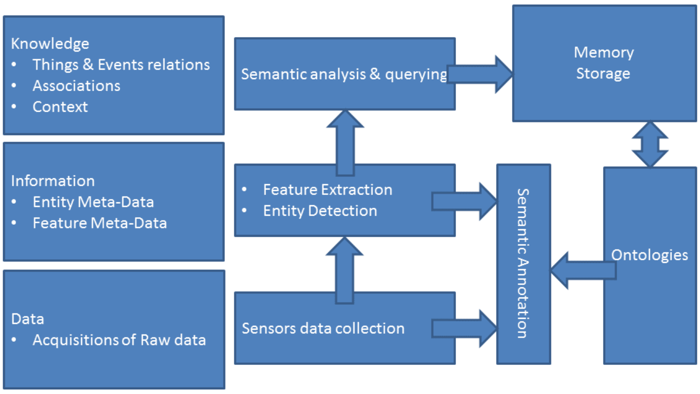


Figure-2.18. The generic data-to-knowledge architectural building blocks.

2.5.6 Home Care

It is a healthcare paradigm facilitating the partnership among medical professionals, individuals, families, authorities, and the market promoting the individual’s health status at home. It is referred as a Long Term Care model distinguished into:

1) **Formal Home Care:** Models that require the aid of paid care and they are distinguished into (i) Supportive Housing, (ii) Independent living, (iii) Assisted living, (iv) Nursing Home, (v) Chronic Care, (vi) Acute Care, and (vii) Hospice Care models.
2) **Informal care:** The model that misses standardized procedures for healthcare services provided to a homebound individual. The Informal Home Care model can potentially be an alternative to institutionalization since it can provide analogous services to Formal Home Care with the same results, given the systematic participation of medical professionals and care givers. Also, it refers to informal long-term care since it involves free of charge services for families and the community.

The effectiveness of the Informal Home Care is determined by the attributes of accessibility, continuity of care, coordination of services, cultural competence, and completeness of services. The model involves the additional roles of (i) Housekeepers, (ii) Homemakers, and (iii) Home Health Aide. The doctors’ visits are classified into four categories due to (a) patient’s incapability to use transportation means, (b) the family concerns, (c) an accident or changes in the behavior, and (d) changes in the individual’s clinical status. The medical doctors visit the patient’s place for (1) a clinical assessment, (2) a follow-up, (c) an illness crisis, or (d) death.

The advantages of the model involve the improvement of the medical care because of the doctors presence at home, which allows to reveal unknown health care needs and possible illness causes. The physician's suggestions consider the social, cultural, economic, and religious issues, enhancing the individual's satisfaction. The disadvantages of home care refer to four parameters: (i) the time intensiveness since a significant part of time is spent commuting between homes, (ii) the limited biomedical support, (iii) the lack of reimbursement procedures, and (iv) the providers' safety, due to the involved risks in the uncontrollable home environment. The observed issues focus on: (i) Logistics, (ii) Information and Communication Technology, and (iii) Culture.

2.6 Home Care Models

Home care is the informal part of Primary Care of the Health System as shown in Figure-2.19, addressing to the entire population, independently of age and gender.

The literature about Home Care put an emphasis on the provision of acute care at home due to the challenges to motivate teams of professionals to perform in the limited standardization of home. The models referred in the related literature include:

1) **Hospital-At-Home (HaH):** a substitutive model of the formal hospitalization.
2) Hospital in the Home and Acute Home-based care.
3) **Chronic Care Model (CCM):** refers to patients suffering from chronic diseases.
4) **Guided Care Model:** relies on the patients' ability to self-management.
5) **Consumer-Directed Services At Home:** refers to people with disabilities.
6) Hospital Elder Life Program (HELP) Model: provision of nursing services.
7) **Acute Care for the Elderly (ACE) Model:** refers to the acute care of the elderly.
8) Program of All-Inclusive Care for the Elderly (PACE) Model.
9) **Palliative Care Model:** reduction of pain and restoration of quality of life.
10) The **TLC** Model of Palliative Care For Older Patients.
11) **Transitions Care Model:** refers to planned medical care recovering at home.
12) Post-Acute Home Care Model in Canada: provision of continuous care.
13) **Chronic Care in Acute Care Setting Innovative Model:** cooperation between the regional hospitals and Home Care.

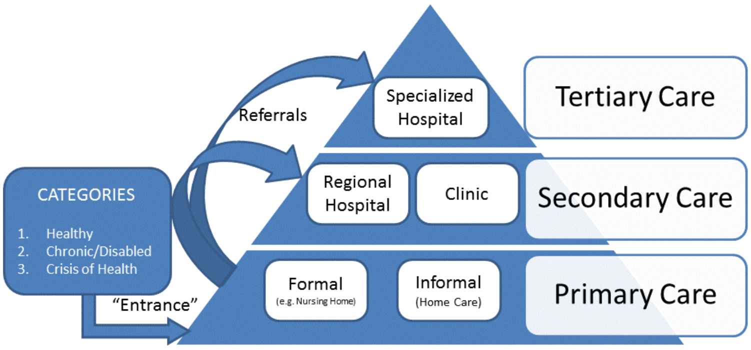


Figure-2.19 Stratification of Healthcare System.

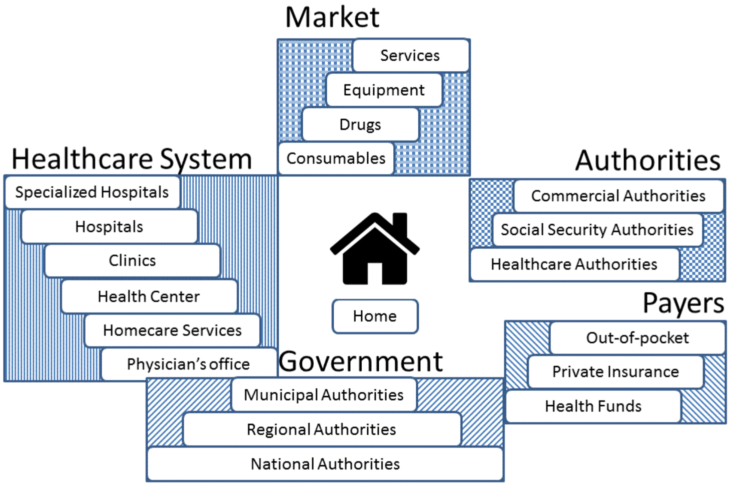


Figure-2.20 The architectural view of the Home Care model.

The above Home Care models reveal the following characteristics and demands.

- **Common Characteristics:** Typical attributes common to Home Care model include: (i) multi-disciplinary teams of professionals adapted to social and cultural needs, (ii) considering safety issues for patients and professionals, (iii) aiming to improve quality of life, (iv) providing independence of living and mobility, (v) providing care, (vi) diagnosis, and prognosis, and (vii) considering the patient's satisfaction from collaboratively offered services and cooperating within a Healthcare System's business model.
- **Specific demands:** The issues related to specific classes of the population, disease specific healing treatments and protocols, and the patients empowerment.

2.7 Home Care Formal Design Approaches

The formal description tackles and excludes conflicts and contradictions including:

1. Functional requirements that describe the objectives, the services, the users, and the observed functionality with high-level statements of the system's performance.
2. Non-functional requirements that describe properties, the structure, and the constraints about the services, organization, and the external requirements.
3. Domain requirements that describe the model's characteristic features that reflect the healthcare domain requirements otherwise the model turns to be useless.

The functional characteristics must show discrete concepts that exhibit operational capabilities and interactions among the model's components and the environment.

2.7.1 Home Care Functionality

The functionality can be expressed either by extensive textual descriptions or systematized tabular forms, or graphical sketches of cellular automata of finite state machines, or using Unified Modeling Language (UML) schemata (www.uml.org) of the Object Management Group (www.omg.org). Each of the above approaches presents limitations and disadvantages related to the quality of the descriptions and the representational completeness of the presenting subject.

In particular, the textual descriptions are tedious and hard to follow when they are lengthy. Similarly, the use of tables to systematically present the functions of a model can be very extensive since it must include, for each function of the model, the description of each task, the objective goal for each task, and the participating actors along with the functionality's pre- and post- conditions. The systematic use of tabular forms requires further to analyze the input, the output, and the processing steps where the participants take part. However, the designer is left unsupported about the formality and the completeness of the provided description. Analogously, the use of cellular automata provides a snapshot of the occurring design showing its incapability to present the dynamic aspects of functionality. The formality of the provided approaches is limited by the two factors, namely the level of detail related to the recipient of the definitions, and the limitations of the descriptive and parametric definitions. UML presents standardized graphical means to express the functional aspects of models employing five key graphical notations: Use Cases, Class Diagrams, Sequence Diagrams, State Chart Diagrams, and Activity Diagrams.

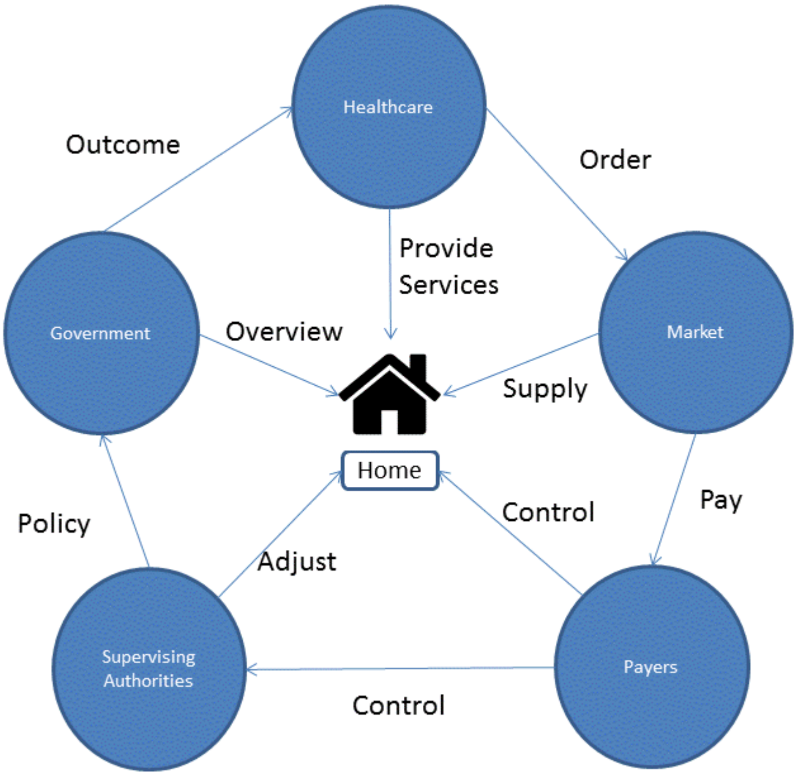


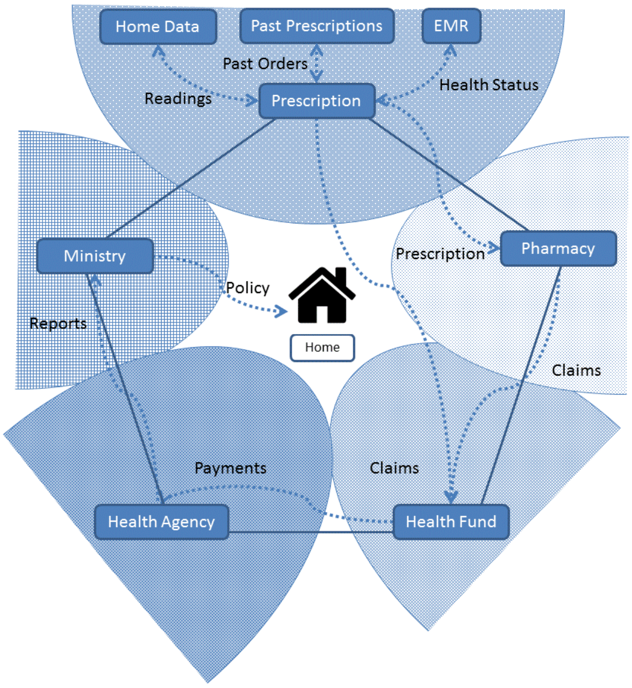
Figure-2.21 The model’s components interactions and interoperation.

The Use Case diagram presents the relationships among users and objects, the Class Diagrams provide the interrelationships among the designed classes of objects, the Sequence Diagrams depict the sequence of actions and instances of classes involved in the design, the State Chart Diagrams provide the discrete states of the involved classes, and the Activity Diagrams present graphically the operations performed on the designed objects describing transitions along with synchronizations and concurrency in the describing workflows. Hence, UML provides the dynamic model with Sequence, State Chart, and Activity diagrams analogously to finite state machine designs missing the needed formality.

2.7.2 Home Care Prerequisites

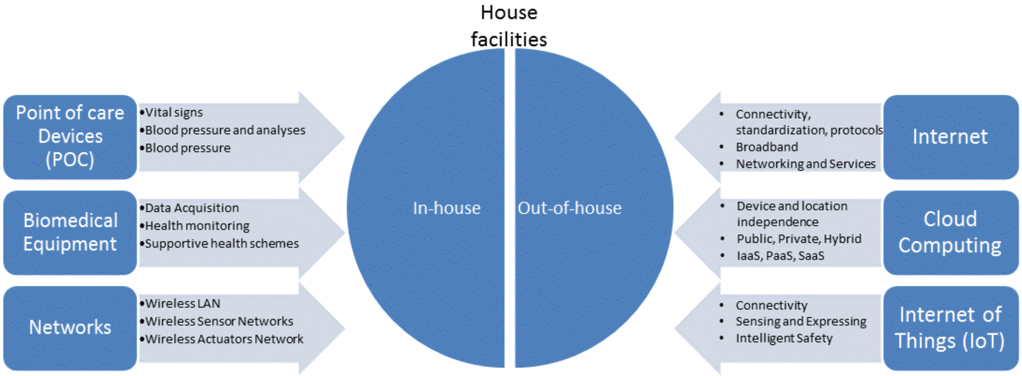
The model’s functionality requires the definition of the involved data structures and the interaction among actors, entities, and concepts with respect to

1) Architecture: An abstract overview of the architectural prerequisites is depicted in Figure-2.20 presenting the major involved components.



2)

3) **Figure-2.22** Example of entities interactions of Home Care model.



4)

5) **Figure-2.23** The house premises infrastructure.

6) The interactions and interoperation: It refers to flows of abstract instructions among the stakeholders as illustrated in Figure-2.21 forming loops ensuring the model's sustainability.

7) Entities interaction: It refers to the interactions and interoperations of the constituting components. Continuing the previous example, the entities participating in the issuance of a prescription are presented in Figure-2.22.

8) Protocols: It describes the interactions among the components and the elements with varying level of abstraction depending on the recipients. Figure-2.23 depicts the protocol by specifying the order of the activities that take place in prescription.

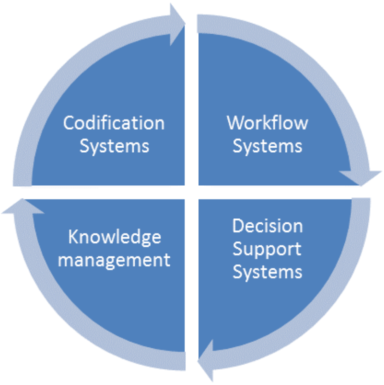


Figure-2.24 Directions for standardizing the provision of medical services at home.

9) Data Structures: It represent the informational structures used in the description of the static operation of the employing software applications that use software routines deposit data and perform interactions among data structures.
10) Data Representations: It is employed to make references to information creating user defined data types augmenting the semantic capabilities of the model.

Hence, the design of the Home Care model must take under consideration the architectural influences, the developing interactions among the identified parameters, and the applying algorithms along with the corresponding data structures.

2.8 The required supporting Infrastructure

The Home Care model's must be viewed with respect to the following properties, (i) the in- and out- of house infrastructure, (ii) the necessary medical infrastructure, (iii) the related operational business requirements, and (iv) the characteristics of governance.

i. In-house / Out-of-house Infrastructure: The in-house infrastructure includes ad-hoc networked and interacting mobile devices. The out-of-house infrastructural complements the in-house one for the completion of workflows.
ii. Medical infrastructure: The set of all conceptual tools and instruments resolving issues related to medical procedures, treatments, diagnoses, codifications, and consumables.
iii. Business requirements: The required processes that allow the functioning of the Healthcare Market.

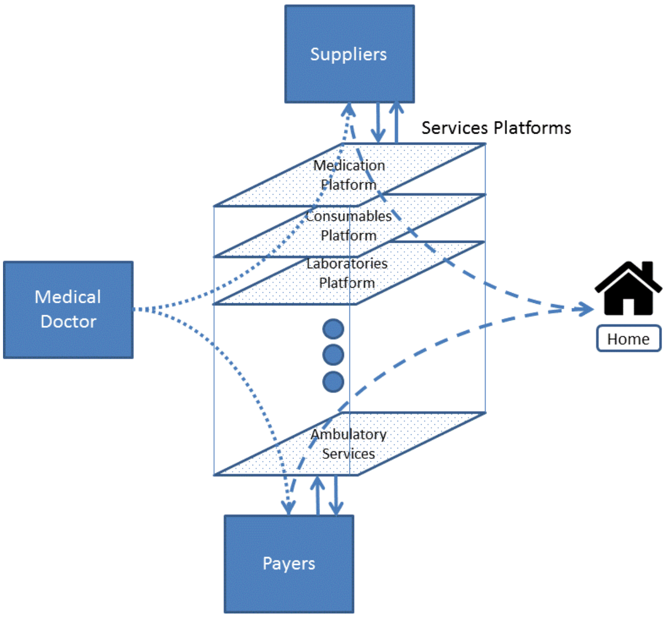


Figure-2.25 Sustainability support in business operations of the Home Care model.

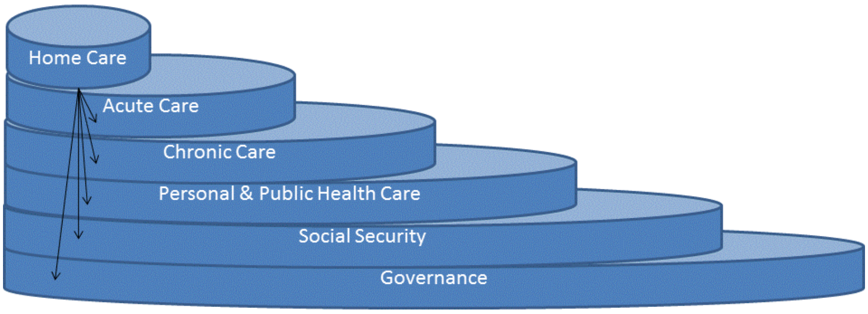


Figure-2.26 A conceptual infrastructure of the Home Care model.

iv. Governance: The interventions through legislation and the follow of operations monitoring performed by State’s Agency to preserve the operational efficiency.

The provision of medical services at home requires the wide spread of exhaustive standardization including codification, workflow, knowledge administering, and decision support systems. Figure-2.24 depicts the standardization requirements of the provision of medical services of Home Care. The infrastructure that supports the Home Care model must provide the capabilities to the representative stakeholders to get involved supporting the provision of services at home. Figure-2.25 depicts the platform implementing the Home Care model’s sustainability. The Home Care model involves governmental concerns which focus on two major

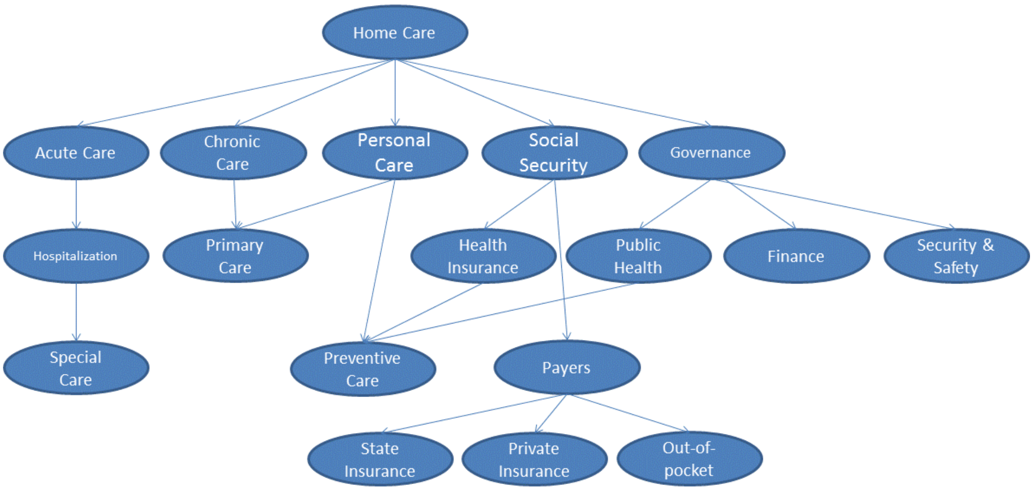


Figure-2.27 A conceptual structure of the Home Care model.

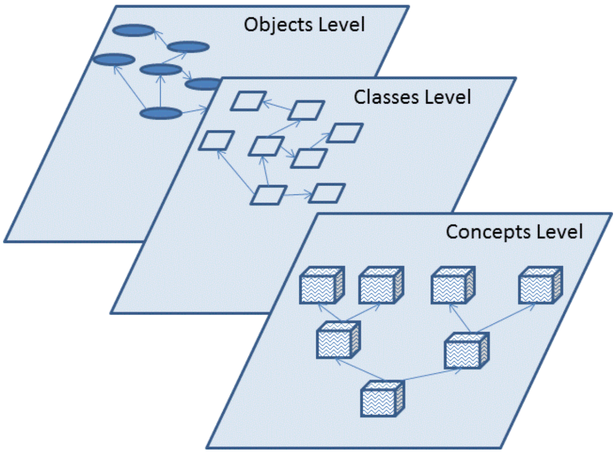


Figure-2.28 A layered structure to perform conceptual references to objects.

characteristics: first, the establishment of a proper legal framework, and second, the establishment of organizations overviewing the application of strategies and policies. Government's presence and interventions are crucial for the model's implementation.

2.9 A Conceptual Model

The Home Care model is consisted of individually interrelated and interacting concepts forming a conceptual structure. The interrelations represent the static behavior of the model while the interactions among the infrastructure's components represent the dynamic behavior. Figure-2.26 depicts the model's structure of the involved concepts.

The Home Care model can be represented by a tree of conceptual interdependencies as sketched in Figure-2.27. The dependencies' manipulations require conceptual tools that allow the composition and tackling complex conceptual structures. The performed operations involve Concept Algebra [2.166] and System Algebra [2.167] provided by Denotational Mathematics [2.168] which is an alternative to analytical mathematics. According to [2.169] there are (i) relational tools performing relation, consistency, and equivalence, and (ii) compositional tools synthesizing concepts involving inheritance, tailoring, extension, substitution, composition, decomposition, aggregation, specification, and instantiation.

The Home Care model is called to express the static and the dynamic behaviors of the architectural components which can be described with a set of classes encapsulating conceptual attributes and properties. Figure-2.28 depicts a layered structure mapping concepts and objects supported by autonomous software mechanisms. Denotational Mathematics manipulates concepts through causal relationships referencing objects.

Synopsis

The choice of the Social Security System model determines the necessary reimbursement scheme for the support of the applying Health System. The formally defined services of the Health System are distinguished into three major layers as illustrated in Figure-2.9. The lowest layer concerns the Primary Care which is considered as the entry point to the Health System as depicted in Figure-2.19 and it is distinguished into the formal and the informal sections. The formal part of Primary Care refers to developed and followed models applying completely controllable medical processes keeping the health related records at the places where care is provided such as in health centers, medical offices, and diagnostic centers. The informal part of Primary Care refers to Home Care with the potential inability to medically control the holding conditions, the carried processes, the participation of medical professionals, and the recording of medical data. The introduction of Ubiquitous Computing in Home Care resolves medically raised issues concerning the requirements of Primary Care and it is expected to face efficiently and effectively parts of Secondary Care substituting, at least initially, specific hospitalized treatments.

Home UbiHealth refers to the provision of healthcare services at home within a ubiquitous computing environment requires a formally defined model to overcome the noticed barriers and raised issues. The model must satisfy the medical requirements supported by an adequate and adjustable infrastructure to facilitate the desired functionality promoting treatments’ personalization, safety for patients and medical professionals, and proper collaboration with the rest of the Health System’s components and the related Market. Determining the formal model involves the formal description of the formed system consisted of heterogeneous components taking advantage of System Theory principles, the dynamically changing processes along with the associated medical values and concepts using an adequately developed algebraic framework for conceptual manipulations, and the static and dynamically changing behaviors of the describing system employing appropriate advanced mathematical devices and tools such as Denotational Mathematics. The necessary mathematical framework must sufficiently accommodate the formal description of the involved clinical data, the performing medical procedures, the carried health related processes, the manipulation of medical concepts, and the administration of the developing medical cognitive systems. Moreover, such a mathematical framework must provide the tools to describe the static and dynamic behaviors in order to administer both intelligence and cognition.

CHAPTER 3. BACKGROUND IN DENOTATIONAL MATHEMATICS

Denotational Mathematics constitutes a formal approach for challenging problems related to Cognitive Informatics and Computational Intelligence [3.1]. Such problems are characterized by large, complex, compound, and heterogeneous in structure architectures with diverging computing behaviors. The observed behaviors can be classified into three categories according to: (i) the acquired property due to the underlying infrastructure characterized by mathematical logic or the "to be" property [3.1], (ii) the enumeration of the sets of attributes or the "to have" property [3.1], and (iii) the functionality performed by the constituents interactions or the "to do" property [3.1].

The behavior is defined as the instantiation of the combinations of operations on memory locations [3.1]. The manipulation of complex problems requires the employment of new mathematical approaches provided by Denotational Mathematics to approach structures containing the operations of recursion, iteration, embodiment of operations, dynamic relations with respect to artificial and physical dimensions, inference, and autonomy. Denotational Mathematics provide the framework to use structured tools that allow the manipulation and administration of knowledge and natural intelligence such as Concept Algebra, Real-Time Process Algebra, and Systems Algebra.

The behavior of software contains both the developer's static and the user's dynamic points of view too. The developer is concerned with structures to obtain the desired output assuming the occurrence of interactions to eventually reach the goals. Formal descriptions of the behavior of software can be expressed [3.1] with the carrying procedures, execution time, and memory space defined as

$$Software\ Behavior = B = f(procedure, time, memory) \qquad \text{(Eq-3.1)}$$

It follows the examination of the formal framework defined by Denotational Mathematics and the associated algebraic tools.

3.1 Requirements

The design of models presupposes the definition of the environment, called the hyperstructure, where the involved principles of concepts, systems, and processes take place.

Definition-1: *Hyperstructure*, $\mathbb{H}$, is the domain [3.2] of mathematical objects represented by tuples of multiple interrelating entries of attributes, properties, and constraints as it follows from [3.2],

$$\mathbb{H} \triangleq \mathfrak{T} = R_{i=1}^{n} F_i(A_i, \Xi_i) = R_{i=1}^{n} < A_i \mid \Xi_i >$$
$$=< A_1| \Xi_1 >< A_2 \mid \Xi_2 > \cdots < A_n \mid \Xi_n > \qquad \text{(Eq-3.2)}$$

with, $A_i$is the i-th attribute in the *hyperstructure* $\mathbb{H}$, $i \in \mathbb{N}$; Ξ_i: i-th applied constraint on A_i, and $R_{i=1}^{n}$: the big-R notation [3.2], with recurring and iterative functionality.

Definition-2: The *concept* C in a *semantic environment* $\mathfrak{U}_c$, with $C \subseteq \mathfrak{U}_c$ and its complement C' be the set of surrounding concepts, is represented by [3.2],

$$C \triangleq (O, A, R^c, R^i, R^o) \qquad \text{(Eq-3.3)}$$

with, O: a finite set of objects of concept, A: a finite set of conceptual attributes and A' those of the surrounding, R^c: a finite set of relations defined by $R^c = O$xA, R^i*: a finite set of input relations defined by* $R^i = A$xA'*, and* R^o*: a finite set of output relations defined by* $R^o = C$xC'

Definition-3: the *system* S in a *system environment* $\mathfrak{U}_s$, with $S \subseteq \mathfrak{U}_s$, is represented by a tuple [3.2] as a closed or open system interacting with its environment,

$$S \triangleq (C, R^c, R^i, R^o, B, \Omega, \Theta) \qquad \text{(Eq-3.4)}$$

with, C: a finite set of components of S and C' be the surrounding components, R^c: a finite set of internal relations among components defined $R^c = C$xC, R^i*:* a finite set of input relations defined by $R^i = C$xC$'$, R^o*:* a finite set of output relations defined by $R^o = C$xC', *B:* a finite set of behaviors defined by $B \subseteq \mathfrak{U}_s$, *Ω*: a finite set of constraints applied on components C, the behaviors B, and relations R defined by $\Omega \subseteq \mathfrak{U}_s$, and *Θ:* a finite set of components outside of the system defined: $\Theta = C' \subseteq \mathfrak{U}_s$

Definition-4: the *process* P in *process environment* $\mathfrak{U}_P$, with $P \subseteq \mathbb{P} \subseteq \mathfrak{U}_P$, is composed by embedded relations [3.2] among n meta-processes following relational rules γ,

$$P \triangleq R_{i=1}^{n-1}(p_i\gamma_{ij}p_j) = (\dots.(((p_1)\gamma_{12}p_2)\gamma_{23}p_3)\dots p_{n-1}\gamma_{(n-1)n}p_n) \qquad \text{(Eq-3.5)}$$

with $p_i, p_j \in \mathbb{P} \wedge \gamma_{ij} \in \mathbb{R} \wedge i = 1, \dots, n \wedge j = i + 1 \wedge i, j \in \mathbb{N}$, big-R functions composition [3.2].

The definitions develop models within the facilitating *hyperstructure* of interacting concepts, systems, and processes integrated and leading to the UbiHealth model.

3.1.1 Concept Algebra

Concept Algebra is the framework within hyperstructures, which facilitates the formal manipulation of abstract concepts by applying classes of operations [3.2].

Definition-5: *Concept Algebra* (CA) is a mathematical framework facilitating the manipulation of concepts applying operators within the semantic environment:

$$CA \triangleq (C, \cdot, \mathfrak{U}_c) = \left((O, A, R^c, R^i, R^o), (\cdot_r, \cdot_p, \cdot_c), \mathfrak{U}_c \right) \qquad \text{(Eq-3.6)}$$

with the term (O, A, R^c, R^i, R^o) representing the formal definition of concept C; the term $\cdot = (\cdot_r, \cdot_p, \cdot_c)$ is the set of used operators classified into $\cdot_r$, the relational operators, $\cdot_p$,the productive operators, and $\cdot_c$,the compositional/decompositional operators; $\mathfrak{U}_c$ represents the universal semantic environment.

The operations performed on concepts can be classified into three classes:

Class-1: *relational* operators in CA

The relational operators perform correlations, associations, and comparisons among concepts which co-exist semantically in $\mathfrak{U}_c$. The set of relational operators is

$$\cdot_r \triangleq \{\leftrightarrow, \nleftrightarrow, \prec, \succ, =, \neq, \sim\} \qquad \text{(Eq-3.7)}$$

with $\leftrightarrow$: relates concepts, $\nleftrightarrow$: denotes the independence of concepts, $\prec$: applies the relation of a concept and its sub-concept, $\succ$: performs the relation of concept and its super-concept, $=$: represents the equivalence between concepts, $\neq$: defines the unevenness and the in-equivalence of concepts, $\sim$: for the relevance between concepts.

Class-2: *reproductive* operators in CA

The reproductive operators perform the creation, the spawning or the reproduction of concepts within the semantic environment$\mathfrak{U}_c$. The set of reproductive relations is

$$\cdot_p \triangleq \{\Rightarrow, \Rightarrow^{+}, \Rightarrow^{-}, \Rightarrow^{\sim}, \longmapsto\} \qquad \text{(Eq-3.8)}$$

with $\Longrightarrow$: used to achieve the inheritance from a concept, $\Longrightarrow^{+}$: achieves the extension of a concept, $\Longrightarrow^{-}$: performs the deduction of a concept, $\Longrightarrow^{\sim}$: applies the replacement of a concept, $\longmapsto$: performs an instantiation of a concept.

Class-3: *composition/decomposition* operators in CA

The set of operators required to perform composition of composite concepts from simple ones or decomposition of complicated concepts to simpler ones is

$$\cdot_{c} \triangleq \{\uplus, \pitchfork, \Lleftarrow, \vdash\} \qquad \text{(Eq-3.9)}$$

with $\uplus$: represents the composition of concepts, $\pitchfork$: the decomposition of a compound concept, $\Lleftarrow$: conceptual ensemble, and $\vdash$: applies particular characteristics to concepts.

3.1.2 System Algebra

System Algebra is the mathematical framework manipulating information that applies on systems modeling and management [3.2]:

Definition-6: *System Algebra* (SA) is a mathematical framework facilitating systems manipulation by applying operators within the formed semantic environment $\mathfrak{U}_s$, determined as follows

$$CA \triangleq (S,\cdot,\mathfrak{U}_s) = \left(\left(C, R^c, R^i, R^o, B, \mathit{\Omega}, \mathit{\Theta}\right), \left(\cdot_r, \cdot_p, \cdot_c\right), \mathfrak{U}_s\right) \qquad \text{(Eq-3.10)}$$

where, the term $\left(C, R^c, R^i, R^o, B, \mathit{\Omega}, \mathit{\Theta}\right)$: represents the systems Definition-3; $\left(\cdot_r, \cdot_p, \cdot_c\right)$: represents the relational, reproductive, and composing/decomposing operators analogously to the operations described in the definition of Context Algebra, and $\mathfrak{U}_s$: represents the systems semantic environment.

The operations performed on systems can be classified into the following classes:

Class-4: *relational* operators in SA

The set of relational operators perform correlations, associations, and comparisons among systems which co-exist in the semantic environment$\mathfrak{U}_c$ is

$$\cdot_r \triangleq \{\leftrightarrow, \nleftrightarrow, \Pi, =, \neq, \subseteq, \supseteq\} \qquad \text{(Eq-3.11)}$$

with $\leftrightarrow$: relates systems, $\nleftrightarrow$: denotes the independence of systems, Π: represents the overlapping behavior of systems, $=$: represents the equivalence between systems, $\neq$:

defines the unevenness and the in-equivalence of systems, $\subseteq$: represents the relation of sub-systems, $\supseteq$: represents the relation of super-system.

Class-5: *reproductive* operators in SA

The reproductive operators perform the creation, the spawning or the reproduction of systems within the semantic environment$\mathfrak{U}_c$. The set of reproductive relations is

$$\cdot_p \triangleq \{\Rightarrow, \Rightarrow^+, \Rightarrow^-, \Rightarrow^\sim\} \qquad \text{(Eq-3.12)}$$

with $\Rightarrow$: used to achieve the inheritance from a system, $\Rightarrow^+$: achieves the extension of a system, $\Rightarrow^-$: performs the deduction of a system, and $\Rightarrow^\sim$: applies the replacement of a system.

Class-6: *composition/decomposition* operators in SA

The set of operators required to perform the composition of complex systems from simple ones or the decomposition of complex systems into simpler ones is

$$\cdot_c \triangleq \{\uplus, \pitchfork, \Lleftarrow, \vdash\} \qquad \text{(Eq-3.13)}$$

with $\uplus$: compose systems, $\pitchfork$: decompose compound systems, $\Lleftarrow$: ensemble systems, and $\vdash$: apply particular characteristics to systems.

3.1.3 Process Algebra

The Real-Time Process Algebra (RTPA) is a framework [3.2] within which can be modelled data objects as Structure Models (SM) and the behavior of systems of data objects as Process Models (PM). Then, RTPA can manipulate abstract SMs and PMs to administer the behavior of processes [3.2] and it is defined as follows:

Definition-7: *Real-Time Process Algebra* (RTPA) is a mathematical framework facilitating the manipulation of meta-process models of relating computational objects within the surrounding semantic environment $\mathfrak{U}_p$ determined by

$$RTPA \triangleq (\mathbb{T}, \mathbb{P}, \mathbb{R}, \mathfrak{U}_p) \qquad \text{(Eq-3.14)}$$

where $\mathbb{T}$ is the defined set of primitive types for modeling computational objects with

$$\mathbb{T} \triangleq (\mathbf{N}, \mathbf{Z}, \mathbf{R}, \mathbf{S}, \mathbf{BL}, \mathbf{B}, \mathbf{H}, \mathbf{P}, \mathbf{TI}, \mathbf{D}, \mathbf{DT}, \mathbf{RT}, \mathbf{ST}, @e\mathbf{S}, @t\mathbf{TM}, @int \odot, s\mathbf{BL}) \qquad \text{(Eq-6.15)}$$

corresponding: **N** for natural numbers, **Z** for integers, **R** for real numbers, **S** for strings, **BL** for boolean values, **B** for bytes, **H** for hyper-decimals, **P** for pointers, **TI** for time, **D** for date, **DT** for data-time, **RT** for run-time, **ST** for system type, @e**S** for event, @t**TM** for timing, @int⊙ for interrupt, and s**BL** for system status respectively. ℙ: the defines set of meta-process to be used for modelling systems behaviors where a meta-process is given in [3.2] and provided in the following definition.

Definition-8: a *meta-process* **P** is a [3.2] primitive-atomic computational operation manipulating the involved variables **V** forming structures **S** of type **T**, it is defined as

$$P = V\text{x}T \rightarrow V^0 \mid S\text{x}T \rightarrow S^0 \qquad 3.16)$$

with V^0, S^0 the terminal-evaluated values of variable or a structure respectively [3.2]

Definition-9: the *meta-process operators* of RTPA are defines by

$$\mathbb{P} \triangleq (:=, \Delta, \Rightarrow, \Leftarrow, \not\Leftarrow, >, <, \mid >, \mid <, @, \triangleq, \uparrow, \downarrow, !, \otimes, \boxplus, \ \S\ ) \qquad \text{(Eq-3.17)}$$

corresponding to assignment, evaluation, addressing, memory allocation, memory release, read, write, input, output, timing, duration, increase, decrease, exception detection, skip, stop, and system, respectively [3.2].

The term ℝ is defined as the set of relational and compositional operators on ℙ primitive types for modeling computational objects in order to synthesize complex system behaviors. The process operator R can be a relational or compositional applied on meta-processes P to develop complex processes as

$$P = P \text{ x } P = (\text{S x } T \rightarrow S^0)\ x\ (\text{ S x } T \rightarrow S^0) \qquad \text{(Eq-3.18)}$$

The term $\mathfrak{U}_s$ represents the process semantic environment.

The operations performed on processes can be classified into the following classes,

Class-7: *algebraic operators* of RTPA

The set of relational operators perform correlations, associations, and comparisons among processes which co-exist in the semantic environment $\mathfrak{U}_p$ is

$$\mathbb{R} \triangleq \{\rightarrow, \curvearrowright, |, | \ldots | \ldots, R^*, R^+, R^i, \circlearrowleft, \rightarrowtail, \parallel, ∯ \quad , |||, \gg, ↯, ↳_t, ↳_e, ↳_i\} \qquad \text{(Eq-3.19)}$$

corresponding to sequence, jump, branch, switch, while-loop, repeat-loop, for-loop, recursion, function call, parallel, concurrence, interleave, pipeline, interrupt, time-driven dispatch, event-driven dispatch, and interrupt-driven dispatch, respectively.

3.1.4 Home UbiHealth model Algebra

The Home UbiHealth model is a system using concepts to apply processes to deliver the desired outcomes within the hyperstructure. Thus, within $\mathfrak{U}$ there coexist interrelated semantic environments, $\mathfrak{U}_s$, embedded concepts, $\mathfrak{U}_c$, and performing processes, $\mathfrak{U}_p$, with:

$$\mathfrak{U} = (\mathfrak{U}_s \cup \mathfrak{U}_c \cup \mathfrak{U}_p) \qquad \text{(Eq-3.20)}$$

Moreover, in (Eq-3.4) the term C refers to components which are, in this case, the concepts included in the system. Also, the terms of (Eq-3.4) referring to the developed relations among the system components and the observed behaviors can be substituted by the performing processes. Hence, in UbiHealth model (Eq-3.4) can be revised into

$$S \triangleq (C, P, \Omega, \Theta) \qquad \text{(Eq-3.21)}$$

Hence, the systemic consideration of the model provides the opportunity to employ principles, methods, and tools from System Theory and Control Theory too.

3.2 Pre-Processing stage

The availability of input, output, as well as the types and the ranges of parameters' values as illustrated in Figure-3.1, can determine the underlying architectural infrastructure.

The Home UbiHealth system is defined by its transfer function relating multi-input with multi-output (MIMO) and affected by disturbance, D(t), Figure-3.1.

$$\text{Transfer function: } F(t) = \frac{Y(t)}{U(t)} \qquad \text{(Eq-3.22)}$$

$$\text{State space form: } \dot{x} = Ax + Bu$$
$$y = Cx + Du \qquad \text{(Eq-3.23)}$$

with A,B,C, and D be coefficient matrices dependent on the nature and structure of the underlying system. Systems and Control Theory principles applying on (Eq-3.22) and (Eq-3.23) provide the following tools:

- Controllability: It is about a randomly chosen input to controllably bring the system from an initial to a final state in a finite period of time.
- Observability: It concerns a chosen output to observe the transfer of the system from the current to a final state in a finite period of time.

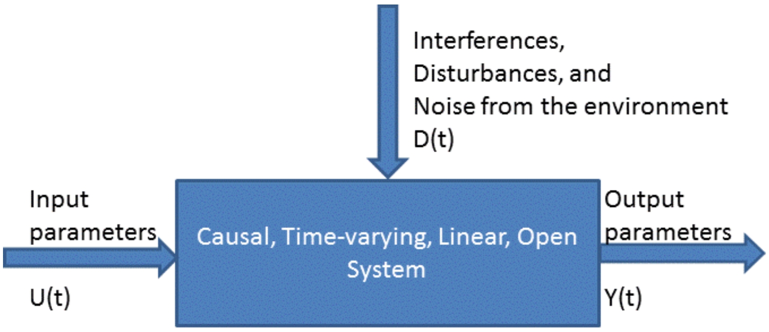


Figure-3.1 Home UbiHealth model approached as an open-system.

- Stabilizability: It specifies whether the system can be brought to a specified state which can be revisited, in a finite number of transitions and period of time.
- Maintainability: It describes the capability to get at the specified state if there is an interaction with the system.
- Stability: It refers to converge from an arbitrary state to a defined equilibrium state, in a finite number of transitions and period of time.
- Detectability: it is about the capability to observe the unstable states or the unobservable states are stable.

The above tools provide the capability to administer the operation of the model.

3.2.1 Types of inputs

The model is a MIMO system with inputs from outside of home, inside of home, the applied medicine, and personal preferences as illustrated in Figure-3.2. The input data are functions of the individual as there can be more than one person treated at home and time.

Input types can be formally described with hierarchical tree structure as we can see in Figure-3.3, uniquely identified, classified, and grouped for fusion to create the dominating values.

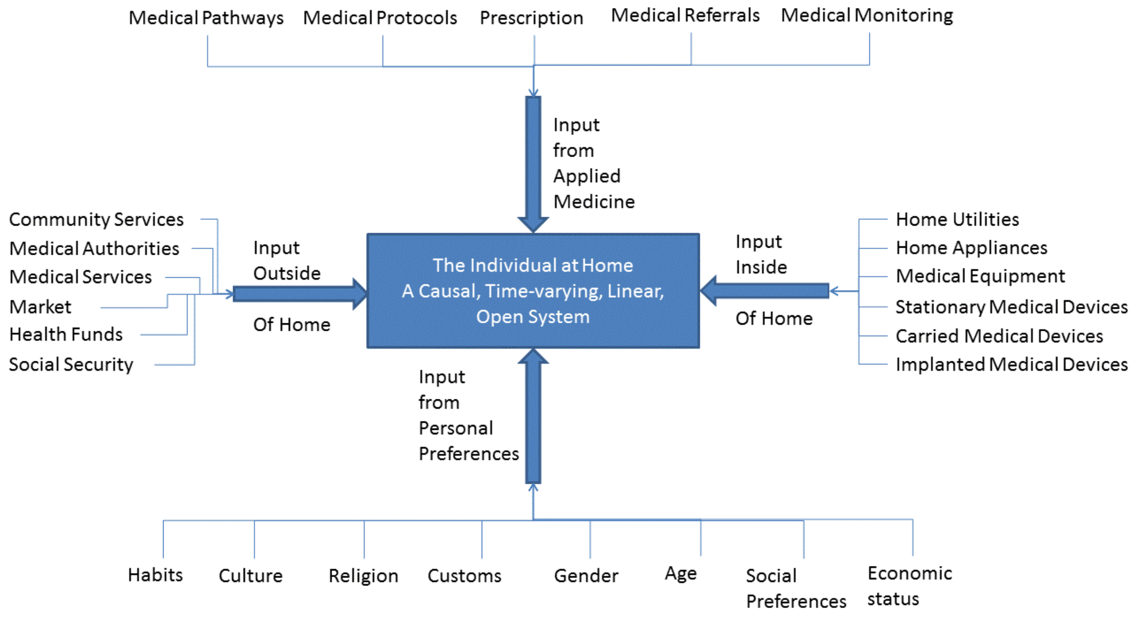


Figure-3.2 Input types feeding the Home UbiHealth model.

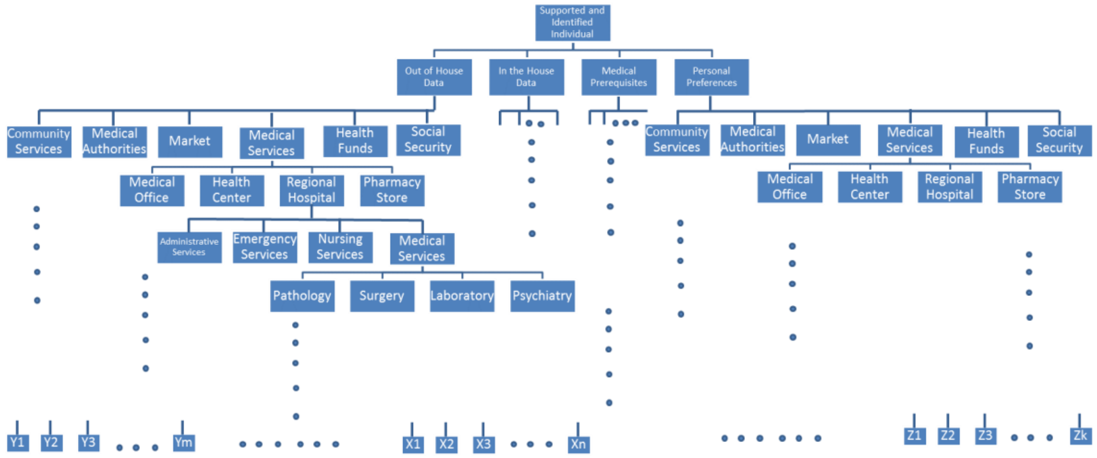


Figure-3.3 Tree structure classifying the input data.

The formal definition requires the definition of a random node of the tree formed in Figure-3.3 and its parent node:

$$node(x, y) = x$$

$$Parent(x, y) = (y, z) \qquad \text{(Eq-3.24)}$$

with $\forall(x, y)\neg\exists(u, v): x = u, y \neq v$ and defining the root node as:

$$individual'sId = root \triangleq node(x, y) = node(parent(x, y)) \qquad \text{(Eq-3.25)}$$

It is required to obtain the path to a particular node and define the root node of the above tree structure. Thus, a path is formed picking a node and recursively reach the root node. Also, the subtree of an arbitrary node x is any node having x in its path. Hence,

$$path(x,y) = \begin{cases} \emptyset & if\ (Eq-3.25)\ holds \\ node(parent(x,y)) \cup path(parent(x,y)) & otherwise \end{cases} \quad \text{(Eq-3.26)}$$

and

$$subtree(x,y) = class\ of\ input = \{(z,u) | x \in (z,u)\} \quad \text{(Eq-3.27)}$$

providing the means to visit the nodes of the tree classifying the input data.

3.2.2 Output outcomes

The World Health Organization (WHO) defines health as the state of physical, mental, and social aspects including well-being beyond diseases [3.3]. Thus, the output refers to the individual's health system characterized by numerous related health parameters that can be measured, compared, used in planning, and adopted in policies [3.4]. Hence, the output is dynamically changing according to the scheme depicted in Figure-3.4, reproducing the Evans and Stoddart framework [3.5] feeding actuators.

The individual's health status can be sensed and classified according to the Evans and Scoddart framework assessing and evaluating illness, risk, or threatening factors. At any time, the health status is a vector containing available output sensors.

$$Output\ Sensors = \{Physical \wedge Artificial\} = S = \{Social, Physical, Genetic, IndividualBehaviors, HealthandFunction, Disease, HealthCare, WellBeiung, Prosperity\} \quad \text{(Eq-3.28)}$$

with S spanning over the linearly independent subspaces of the contained vectors.

3.2.3 System Transitions

Events cause the system to be assigned at one of the states in the State Space, using the state's parameters received from the pool of Parameters Space, and applying constraints. The system transitions represent the adapted behavior as in Figure-3.5.

The transitions follow a typical Markov chain as shown in Figure-3.6 and at each state S at time k, the system identifies the next possible states that may occur when system

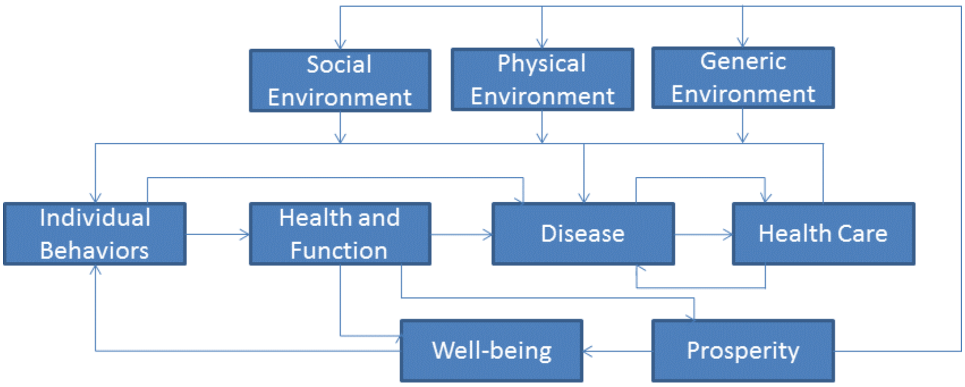


Figure-3.4 The framework of determinants of health [3.5].

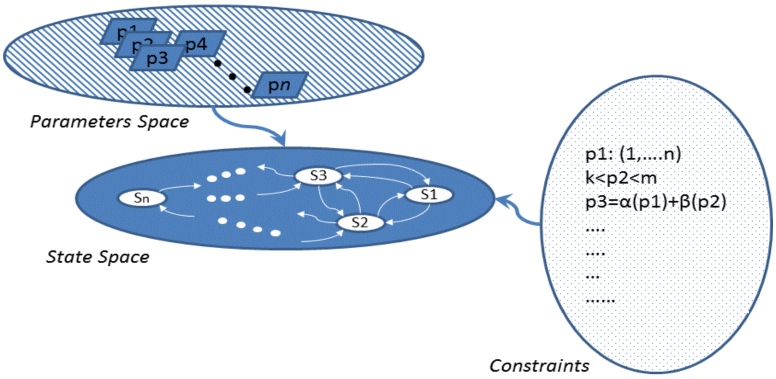


Figure-3.5 The transitions supported by the Parameters Space and Constraints.

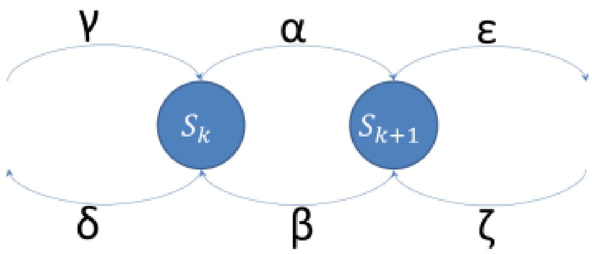


Figure-3.6 A representative Markovian chain with transition probabilities.

is observed at time k+1. The model is described by the state transition matrix A, a square matrix mxm with the relations among the m parameters from the Parameters Space, called state variables. At each state, the sum of flows in and out of a state must be equal to 1. The output can be expressed as a linear combination of the state variables and the input. At state S_k, it holds that,

$$S_k: \alpha + \delta - \beta - \gamma = 1 \qquad \text{(Eq-3.29)}$$

The formal definition of a Markov process is provided by the Chapman-Kolmogorov intermediate state equations given by

$$Pr = \{X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k\} \quad or$$
$$P_{ij}(k, k+n) = \Pr(X_{k+n} = j | X_k = i\} \, or$$
$$p_{ij}(k, k+n) = \sum_{r=1}^{R} p_{ir}(k, u) p_{rj}(u, k+n) \qquad \text{(Eq-3.30)}$$

provided that $\sum_{j \in \Gamma(i)} p_{ij}(k) = 1,$ the state space provides useful definitions and properties for monitoring and controlling the behavior of the developed system with,

- Path: Given any two states i and j, the sequence of transitions from i to j with positive probabilities of occurring.
- Reachable states: It defines the path from state i to reach state j.
- Closed subspace: A subset S of the state space is closed if $P_{ij} = 0$ for $\forall i \in S \wedge j \notin S \wedge i, j \in N$.
- Absorbing/non-absorbing state: It is about the capability to perform transitions originated from a set of system's states to a defined absorbing state. The absorbing state is a single element and it is member of a closed set of states, otherwise the state is considered as non-absorbing.
- Irreducible states: It refers to a closed set S of states, if any state $j \in S$ is reachable from $\forall i \in S \wedge i \in N$.
- Recurrent state: It occurs when the system returns to a state sometime later after leaving from it.
- Periodic/Aperiodic state: It is noticed when a state i is visited after a number of steps.
- Discrete/Continuous transition: The transition can take place among countable states at known discrete time intervals otherwise time is not known and it is considered as continuous.
- Ergodic states: It is about the states characterized by recurrence and aperiodicity. Also, there are encountered positive probabilities connecting the developing-formed state paths.
- Steady-State or equilibrium or stationary state: At each state as seen in Figure-3.6 it holds the equivalence of $Rate\ of\ flow - in = Rate\ of\ flow - out$. Hence, if the input parameters are becoming time-invariant, then a solution can be obtained independently from time in spite of the fact that some model

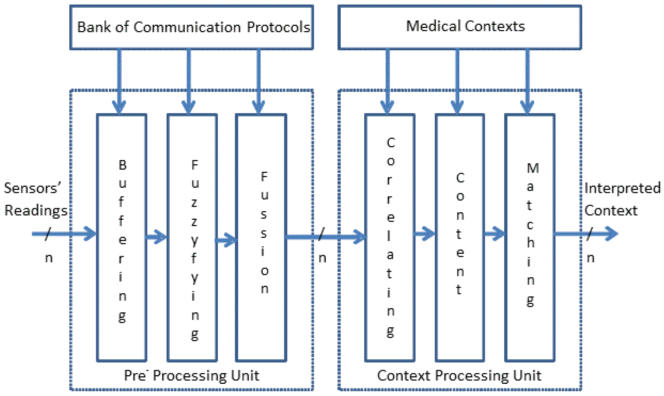


Figure-3.7 A block diagram model of the pre-processing architecture.

elements are described as functions of time. More formally, if a unit step function is provided at the system's input, then the system's steady-state is the system's output value at time T=∞.

The known and applied procedure includes the following steps (1) a rough description or sketch of the system, (2) defining the involved states, (3) defining relations among states, (4) developing the transition matrix, and (5) calculating results. Solving systems of equations is prohibited by the available computing power. Thus, designing the state space follow the steps of enumerating states, identifying transitions, and assigning state variables keeping their number as low as possible.

The transition of the system Tr reflects its behavior and its model is defined by

$$Tr =< S, S_o, R, \alpha, \beta > \qquad \text{(Eq-3.31)}$$

where, S: a finite set of defined states, S_o: the initial values of the defined states, R: set of relations among the states, i.e., the set of transitions, and α, β: two relations from R to S associating the source $\alpha(r)$and the target $\beta(r)$ states. Thus, a transition can be represented by Tr: $s_i \rightarrow s_j$.

The State Space provides defined and related states. The association of states, from $x(t)$ to $\bar{x}(t)$, leads to $\bar{x}(t) = T^{-1}x(t)$ where T is any non-singular matrix called *similarity matrix*. Extending the meaning of similarity, the system can be defined as:

$$\begin{aligned}\dot{\bar{x}}(t) &= \bar{A}\acute{\bar{x}}(t) + \bar{B}\bar{u}(t)\\ \bar{y}(t) &= \bar{C}\bar{x}(t) + \bar{D}\bar{u}(t)\end{aligned} \qquad \text{(Eq-3.32)}$$

where, x: the state vector, u: the control signal, and y: the output

Then, from the similarity relation $\bar{x}(t) = T^{-1}x(t)$ it is obtained:

$$\bar{A} = T^{-1}AT = \Lambda = eigenvector\ of\ A \quad \text{(Eq-3.33)}$$

$$\bar{B} = T^{-1}B \quad \text{(Eq-3.34)}$$

$$\bar{C} = CT \quad \text{(Eq-3.35)}$$

$$\overline{D} = D \quad \text{(Eq-3.36)}$$

The solution of the state equation model can be solved providing:

$$Y(s) = |C(sI-A)^{-1}B + D|U(s) + C(sI-A)^{-1}x(0) \quad \text{(Eq-3.37)}$$

with $x(0)$ to represent the initial conditions and transformed in the Laplacian s-plane. Thus, choosing different state variables leads to different internal descriptions of the model but using still the same input-output model.

The development of the above model provides the opportunity to use the following administrative tools:

- **Controllability:** it can steer the state via the control of the input to certain locations-states in the State Space. Formally, the state x is said to be controllable if there exists a finite interval of time [0, T] and an input $\{u(t), t \in [0,T]\}$ such that $x(T) = 0$. If all states are controllable then the system is said to be completely controllable depending on input. Testing the controllability property is performed by:
  If $\det(sI-A) = s^n + a_{n-1}s^{n-1} + \cdots + a_0$ then, $A^n + a_{n-1}A^{n-1} + \cdots + a_0 I = 0$ or

  $$\Gamma_C[A,B] \triangleq [B\ AB\ A^2B\ \ldots .A^{n-1}B] \quad \text{(Eq-3.38)}$$

  It is completely controllable iff $\Gamma_C[A,B]$ has full row rank.
- **Reachability:** A state $\bar{x} \neq 0$ is reachable from the considered origin if given $x(0) = 0$, $\exists t \in [0,T]$ and input $\{u(t), t \in [o,T]\}$ such that $\{x(T) = \bar{x}$. If all states are reachable, then the system is called completely reachable.
- **Stabilizability:** a State Space model is stabilizable if its uncontrollable subspace is stable.
- **Observability:** it can estimate the state via the control of the output. Formally, a state $\bar{x} \neq 0$ is unobservable if given $x(0) = x_0$ and $u[k] = 0, k \geq 0$ then $y[k] = 0, k \geq 0$.
  A system is completely unobservable if there exists no nonzero initial state that is unobservable. The observability test is performed by:

  $$\Gamma_o[A,C] \triangleq \begin{bmatrix} C \\ CA \\ \ldots \\ CA^{n-1} \end{bmatrix} \quad \text{(Eq-3.39)}$$

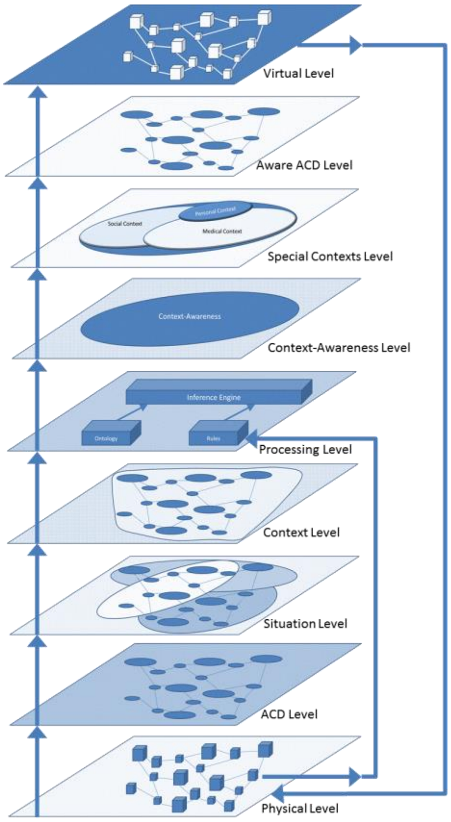


Figure-3.8 The development of context-awareness process.

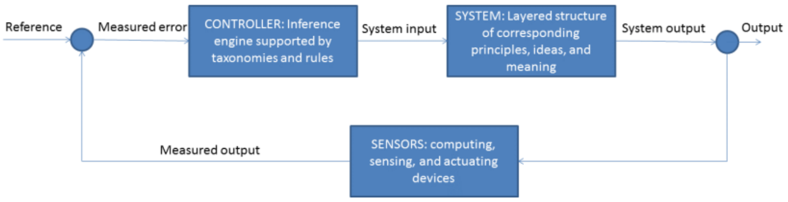

Reference
Measured error
CONTROLLER: Inference engine supported by taxonomies and rules
System input
SYSTEM: Layered structure of corresponding principles, ideas, and meaning
System output
Output
Measured output
SENSORS: computing, sensing, and actuating devices

Figure-3.9 Context-awareness control by a first-order feedback loop.

The system is completely observable if $\Gamma_o[A, C]$ has full column rank.

- **Detectability:** the system model is detectable if its uncontrollable state subspace is stable.
- **Duality:** for a given model represented by the tuple (A, B, C, D), then if (A, B, C, D) is completely controllable then $(A^T B^T C^T D^T)$ is completely observable.
- **PBH Test:** determines the properties of controllability and observability providing:

If and only if there exists a non-zero vector $x \in C^n$and a scalar $\lambda \in C$ then

$$Ax = \lambda x$$
$$\mathrm{C}x = 0 \qquad \text{(Eq-3.40)}$$

Then the model is completely observable.
Similarly,

$$X^T A = \lambda X^T$$
$$X^T B = 0 \tag{Eq-3.40}$$

Then the model is completely controllable.

3.2.4 Pre-Processing Architecture

Figure-3.7 depicts a model with two units for the reception and preparation of raw data. The first unit pre-processes data by (i) storing in buffers the recognized data, (ii) applying fuzzy methods to reduce imprecision, and (iii) fusing data. The coordination is performed by the Bank of Communication Protocols guiding identification and handling of the received data.

The second unit, correlates data to define artificial parameters, finds all matching classified data structures from known medical contexts, and fuses the completeness of formed candidate contexts giving at the output candidate medical contexts to be examined.

3.3 Context-Awareness for Interoperability

Interoperability is a major characteristic prerequisite of the heterogeneous participating devices of the Home UbiHealth model which can achieve interoperation at the semantic level. The semantic interoperability allows the participating devices to share the same meanings and inferences in the forming and evolving situations in the model. Figure-3.8 presents a layered framework with feedback. The interaction among the physical devices occurs *at the physical layer* which provides a data structure for each device or the Application Customized Description (ACD) data structure *to the situation layer*. The relations of ACDs produce *the context layer*. The input of constraints and personal preferences produces *the processing layer*. Additional relations are formed with dedicated taxonomies and inference engines producing *the context-awareness layer*. The distinction of dedicated and overlapping contextual sets produces *the special context layer* such as the medical, the social or the personal contexts. Additional parameters develop for each device producing *the aware ACD layer***.** The use of the domain knowledge produces *the virtual layer* where the interoperation among the participating devices is performed at the virtual level with complete compatibility. Then, the resulting interactions are carried to the physical layer.

The framework depicted in Figure-3.8 includes the following mechanisms: (i) inference engine, (ii) rules system, (iii) taxonomies administrator managing the representation of knowledge, (iv) layers interface, and feedback mechanisms.

From a different perspective, the semantic interoperability framework can be presented as a first order feedback loop as shown in Figure-3.9 with inputs received from the devices. The framework processes concepts using knowledge representations.

3.3.1 Development of Awareness

Context-awareness is obtained by the cooperation of the inference engine with the support from a rule-based system and dedicated taxonomies. The Processing Layer involves the operations:

1. Identifying concepts using taxonomies and rules-based systems.
2. Normalizing concepts by determining the relations among the involved concepts (taxonomies), discovering the inferred relations, and using the ontological axioms.
3. Reasoning applies a framework of quantified rules provided by the physical environment resulting into inferences enriching the context data and leading to context-awareness.

The employed taxonomies serve two major and discrete purposes knowledge representation and knowledge querying. The reasoning provided by the inference engine is available approaching the taxonomies with a query language to access the taxonomy’s coded concepts, to activate the embedded taxonomical relations and axioms, and to apply the quantified constraints provided by sets of rules. The use of the specifically designed Context Ontology Language (CoOL) [3.6] augments the richness of the contextual data and develops the contents of the Context Awareness Layer. The use of CoOL requires the adaptation of the Aspect-Scale-Context (ASC) model described in [5.18] involving the concept of Aspect which participates in one or more Scales and each Scale is contained in one or more Contexts. An Aspect represents conceptual characteristics in the adopted taxonomy which can be contained other more general characteristics of the taxonomy or an unordered set of aspects which, in turn, can be referred to Contexts or an unordered set of Scales [3.6]. The ASC model is accessed with CoOL to develop context-awareness representing knowledge and transfer it to other models [3.6]. At the Processing Layer each operational contextual model can be compared with the stored ones.

3.3.2 Interoperability Issues

The development of the Virtual counterpart of the Physical Layer, Figure-3.8, is the prerequisite for the achievement of semantic interoperability. The availability of the model driven knowledge at the Processing Layer offers the availability of a core or static ontology consisted of specific taxonomies related to the domain of medicine, as well as instances of the Physical Layer. The knowledge representation is similar and analogous to the knowledge representation provided in [3.7] with the distinction between the T-box and A-box for the knowledge representation as it is reproduced in Figure-3.10. which produces a knowledge meta-model in the next layer. The Resource Description Framework (RDF) is adopted from [3.7] using the triple $\{subject, predicate, object\}$ with the object to refer at Physical Layer's contents, the subject refer to the used taxonomies' elements, and predicate refers to rules. The knowledge base contents from [3.7] is adopted using descriptions for the participating devices, their attributes and properties, their current context, their users, their configuration, and their procedural steps [3.7]. The procedural steps or recipes contain sequences of structured steps consisted of terms from the used taxonomies. The matching synchronization between the functionalities and the interfaces among the participating devices is performed by a virtual communications module [3.7] to achieve the production of the interoperability interface of the virtual devices.

3.3.3 Modelling Requirements

The framework depicted in Figure-3.8 and Figure-3.10 provides the conceptual infrastructure for the achievement of semantic interoperability through the adoption of the Recipes [3.7]. The construct of Recipe as a sequence of procedural steps containing instructions to perform conceptual and functional matching is used to adapt virtual devices' interfaces under the constraints of context. Hence, the

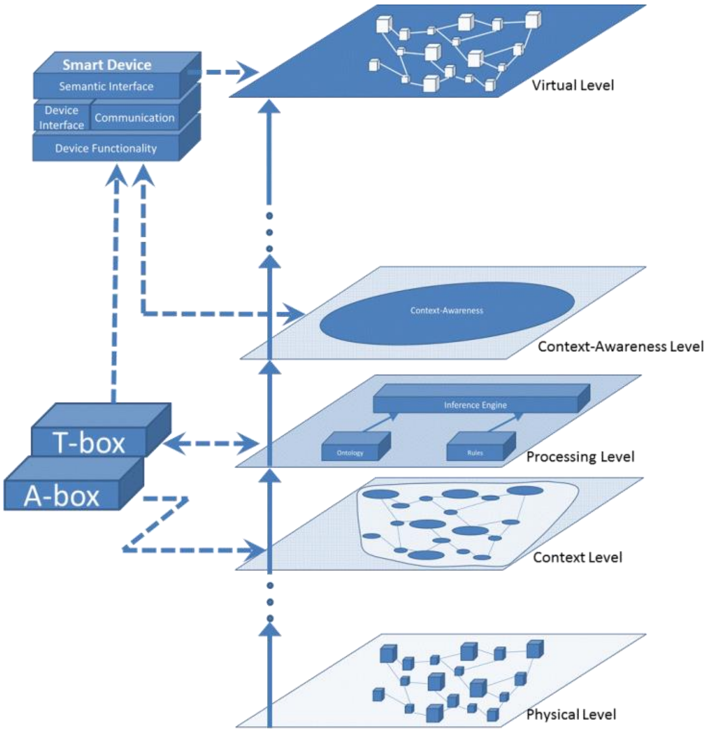


Figure-3.10 Context-aware interoperability of smart devices [3.7].

framework presented provides the captured physical aspects at the Physical Layer and interprets the corresponding artificial aspects developed and give at the Virtual Layer. The interpretation involves capturing of the semantics of the developed relationships among the participating devices and persons which must be described formally and provided back to the Physical Layer. A raw representational model of each of the participating devices in the Physical Layer is provided with each device's description at the ACD Layer. Extending this idea stereotypes can be developed defining the activity and the roles performed by each device providing flexibility in the description, extensibility in the involvement of attributes and properties, reusability describing similar devices, and applicability due to easiness of classifications in conceptual manipulations [3.8]. The model presented in [3.8] uses the personas' technique to describe user archetypes which is adopted and extended to describe every entity in the Physical Layers of our framework since it provides the capability to classify, adapt, customize, relate, and share attributes and properties. The Home UbiHealth model needs the personas' technique [3.8] to administer the large number of participating entities at the Physical Layer using the profile data of each entity both at design and run time too.

The formal description of the interoperability characteristics of the Home UbiHealth requires the definition of the participating entities such as the devices

$D_i$and the users U_j with $i \in N$ the set of natural numbers. The users interact with devices and both of them interact with developed information entities. The information entities are forms of meta-data resulted at the layers of the adopted framework and they are composed of concepts, properties, and variables. The framework uses implementations of limited ontologies O_k, $k \in N$ integrated into knowledge bases which are used to obtain meanings of the administered concepts with direct references to entities, devices and users. The meanings are semantic entities that must be formally defined and the devices and the users have varying interest on them. The following items must be formally defined in order to describe formally the Home UbiHealth model with

- Universe, V, at the physical layer it contains devices D_i, users U_j, and entities E_l.
- Concepts, S, are applied on the universe defined by $\mu_S: V \to \{0,1\}$ with $S = \{S_i\}, i = 1, \dots n$.
- Ontologies, O, defined by $O = \{S, R\} = \{S, SxS\}$ and R are the developed relations.
- Relations, R, of the involved elements with $R_i(x, y)\, with\, x, y \in S$ and $R_i^{-1}(x, y) = R_i(y, x)$. Then the developed relationships on concepts must be considered: Union of concepts, $(R_i \cup R_j)(x, y)$, Intersection of concepts $(R_i \cap R_j)(x, y)$, and Composition of concepts $(R_i{}^\circ R_j)(x, y)$ forming relational functions such as $(R_i \cup R_j)(x, y) = f(R_i(x, y), R_j(x, y))$, $(R_i \cap R_j)(x, y) = g(R_i(x, y), R_j(x, y))$, and $(R_i{}^\circ R_j)(x, y) = \sup f\,(R_i(x, y), R_j(x, y))$.
- Algebraic manipulation of relations processing large volumes of data, for instance, the intersection of relations provides the smallest set of characteristics of the united relations. Hence, sets of operations can be defined and developed to handle algebraically the involved concepts such as:

$$R_P(x, y) = \begin{cases} 1 & \text{concept} - x\; matches \;\text{concept} - y \\ -1 & \text{concept} - x \text{ is } inverse\ to \;\text{concept} - y \\ 0 & \text{concept} - x\; no\ matching\ \ to \;\text{concept} - y \end{cases}$$

From the above relationship among concepts we can determine context with $R = \cup_i \left(R_i^{P_i}\right)$ with $P \in \{0,1\}\; and\; i = 1, \dots n$. The last equation applies on relations providing the capability to build combined conceptual constructs. The consideration of fuzzy techniques to adjust granularity on the administration of concepts, the use of techniques to take advantage of the evidences received from sensors such as the Dempster-Shafer method, and the application of fusion techniques on concepts.

The framework depicted in Figure-3.8 requires a software infrastructure consisted of agents and individual mechanisms. Figure-3.10 presents the dynamic nature of the use of Recipes to achieve semantic interoperability. In the figure below, first, an agent software mechanism performs the correspondence between the elements of the ACD Layer that contains the physical attributes of the participating entities in the Physical Layer and the entities contained in the Context-Awareness Layer. Second, the entities participating in the Context-Awareness Layer export their characteristic variables, values, ranges of values, and functionality to recipes. At the third step, the agent makes the association among the ACD profiles, the contextual entities, and the executing recipes. The Home UbiHealth model requires a static infrastructure capable to accommodate the relationships and the mappings of the involved entities to provide a uniform operational computing environment. Also, forming recipes a mechanism or some language is needed to determine procedures and conceptual mappings.

3.4 Adaptation Mechanisms

The layered framework of the Home UbiHealth model receives input from the physical and induces the virtual layer. The dynamic nature of the model requires software mechanisms to achieve, the feedback of the virtual dependencies and relations that synchronize with the physical layer, and the enforcement of operational rules and constraints from the physical to the processing layer of the framework. Thus, two software mechanisms are responsible to:

1. connect and synchronize the physical with the virtual worlds, and
2. introduce operational constraints and personal preferences.

Figure-3.11 depicts the required adaptation mechanisms along with the rest of the supporting mechanisms applied on the hierarchical structure of the framework. Mechanism-10, is a negative feedback providing contextual-awareness to the physical devices. Mechanism-9 is a positive feedback mechanism that inserts into the layered framework operational constraints and personal preferences. Hence,

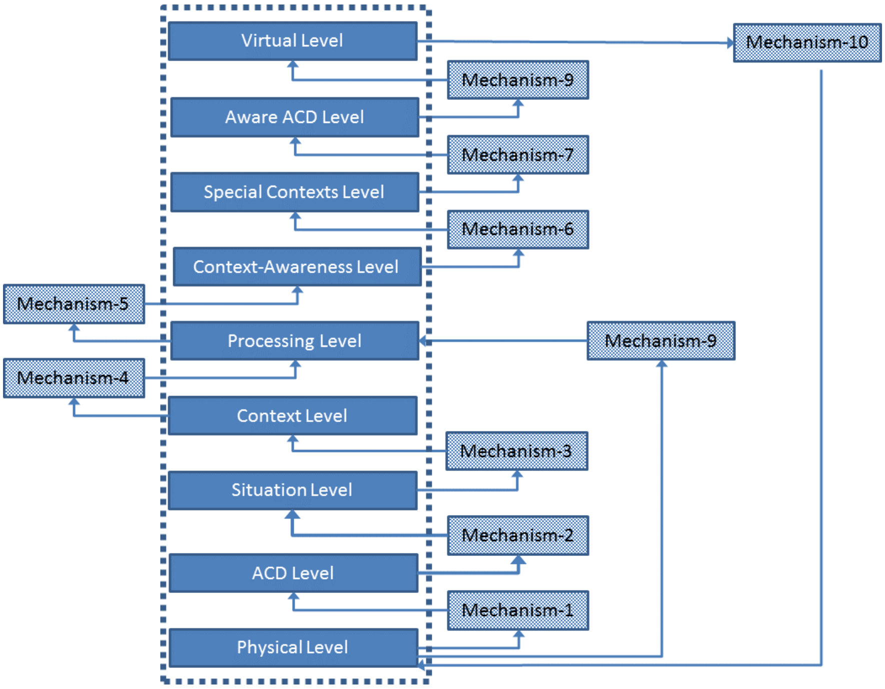


Figure-3.11 The supporting software mechanisms of the context-aware framework.

Mechanism-9 sets constraints and Mechanism-10 tries to gradually meet the aimed targets. The framework operates autonomously without a central control mechanism and without external interventions.

3.5 Processing

The UbiHealth model processes the received data according to the following pattern:

- The physical layer collects the signals from the participating devices and develops the next upwards layer consisted of the participants ACD data structures through **Mechanism-1** forming the ACD Level.
- Then **Mechanism-2** performs comparisons to enrich the data structures with additional information spawned from the holding relations among the available data received from the lower level to form the Situational Layer.
- The received set of data is examined further by inter-relating that data and the resulted artificial data in the Situation Layer obtained by **Mechanism-3** constitutes the contextual data at the Context Layer.
- The contextual data and structures populate the available medical protocols, diseases, therapies, and standards templates to develop by **Mechanism-4**

conceptual candidate prototypes based on available data at the Processing Layer.

- At the Processing Layer, the available medical templates are further processed by the introduced medical constraints and personal preferences by **Mechanism-5**.
- The templates are processed by **Mechanism-6** using ontologies enriching further and making available the data to the Context-Awareness Layer.
- Then, the **Mechanism-7** lead to Special Context Layer where the developed context-awareness is split into domains according to medical knowledge.
- **Mechanism-8** uses the available knowledge and the available data to develop augmented ACD from the available templates forming the Aware ACD Layer.
- The aware ACD structures supported by **Mechanism-9** form the Virtual Layer where virtual entities interact in correspondence to the Physical Layer entities.
- Interactions among the layer's entities, physical and artificial, provide the conceptual interactions and **Mechanism-10** performs the feedback to Physical Layer providing those signals that allow the Physical Level to meet its objectives.

The stack of services receives support from taxonomies, the constraints enter through Mechanism-5 with interfaces, and Mechanism-10 acts as a feedback controller.

3.5.1 Layered structure for sequential linear processing

Each layer transforms the layer's data, formally, performing linear composition according to:

$$f_1(x) = x, f_2(y) = y, f_3(z) = z, f_4(w) = w, \dots, f_9(q) = q \qquad \text{(Eq-3.41)}$$

with

$$F = ((((((((f_1{}^\circ f_2)^\circ f_3)^\circ f_4)^\circ f_5)^\circ f_6)^\circ f_7)^\circ f_8)^\circ f_9) = ((((((((x^\circ y)^\circ z)^\circ w) \dots)^\circ q) \quad \text{(Eq-3.42)}$$

where,

- $(f_1{}^\circ f_2) = f_{12}$ providing the ACD structures
- $(f_{12}{}^\circ f_3) = f_{1..3}$ providing the formed situation
- $(f_{1..3}{}^\circ f_4) = f_{1..4}$ providing the developed context
- $(f_{1..4}{}^\circ f_5) = f_{1..5}$ providing the pre-processed data

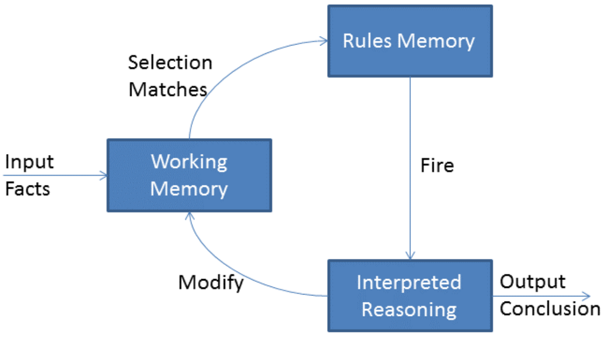


Figure-3.12 A block diagram of the building blocks of a typical rule-based system.

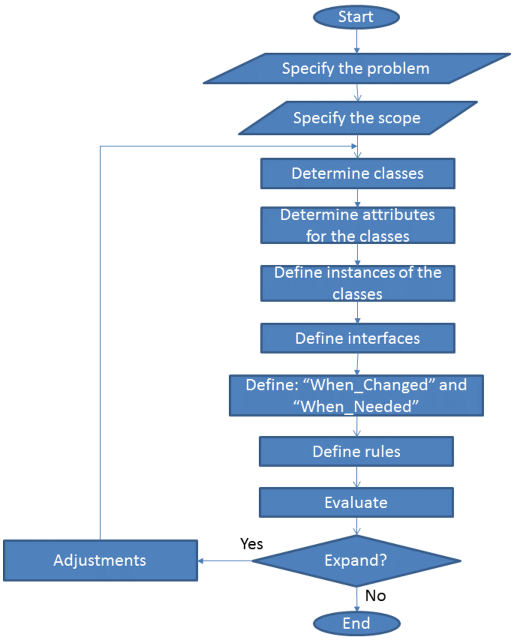


Figure-3.13 Designing steps for a frame-based system.

- $(f_{1..5}{}^{\circ}\, f_6) = f_{1..6}$ providing the awareness to context
- $(f_{1..6}{}^{\circ}\, f_7) = f_{1..7}$ providing the specific contextual characteristics
- $(f_{1..7}{}^{\circ}\, f_8) = f_{1..8}$ providing the aware ACD structures
- $(f_{1..8}{}^{\circ}\, f_9) = f_{1..9}$ providing the virtual level

Each function is a first order polynomial transforming data from one level to another.

3.5.2 Mechanism-5: Introducing Medical Constraints and Personalization

A rule-based system input as depicted in Figure-3.12 is tested against the stored rules. It is interpreted with forward or data-driven reasoning on facts, and proceeds forward with intermediate conclusions until exhausting all data. Also, it is

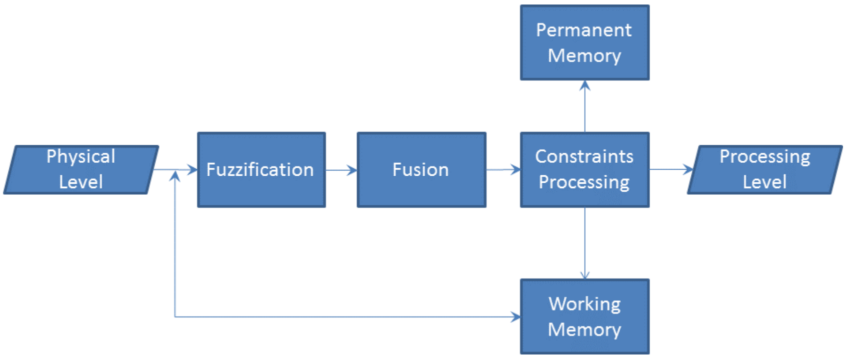


Figure-3.14 Design principles of Mechanism-5.

interpreted with backward or goal-driven, testing a hypothesis and moves backwards examining sub-goals until to prove it.

Frame-systems employed to design classes of constraints consisted of slots using object-orientation. Then, slots of different frames can be referenced and accessed by inference engines using facets which are entities that dictate the use or the process of slots' contents. By inheritance the frames can be part of families of frames achieving generalizations such as "is-a", "kind-of", or "part-of" and associations such as "belongs-to" or "owned by" etc. Thus, they develop generalization, aggregation, association, and inheritance. On frames operate methods and demons. The methods are procedures activated by semaphores signals of "When Changed" or "When Needed" and demons preserve the characteristics of rules (If-Then-Else structures). The "When Changed" signal updates the involved attributes in cases of changes while the "When Needed" searches for the most recent ones. Figure-3.13 shows the flow with the constituents,

1) Classes: With (a) the involved classes of frames (classes of objects) and (b) he attributes of each class and the common attributes among classes.
2) Frames: Defining sets of constraints.
3) Facets: frames' references through defined conditions and (a) values of the frame's slots (default, ranges, and type), (b) interfaces for the user and other frames, and (c) inference process, for instance, the processing can stop when attributes change or a values must be acquired before proceeding processing.
4) Slots: are constituents of frames characterized by attributes that can represent compound attribute–data structures and procedures that they can take advantage of polymorphism. Slots can store knowledge and they can manipulate reproducible knowledge with demons whose functionality is limited to the manipulation of rules and methods that provide procedural

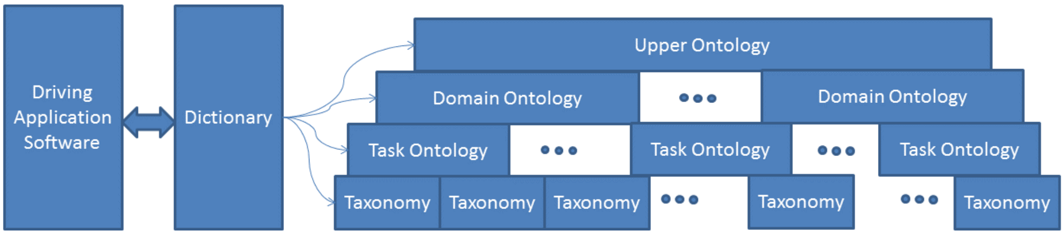


Figure-3.15 The building blocks of the ontology service.

knowledge with semaphores of values of either When Changed or When Needed.

5) The inheritance tree structure along with the relations among the inherited frames and the default values of the inherited attributes.

However, the large number of available sources causes uncertainty requiring the Dempster-Shafer Evidence Theory (DSET) and fuzzification to exclude it. Briefly, the (DSET) considers the set of all possible mutually exclusive events which are assigned the degree of belief of evidence, $\sum_{E\in P(\Theta)} m(E) = 1$. The combination of two or more evidences is provided by the Rule of Combination evaluated by $(m_1 \oplus m_2)(z) = \sum_{X\cap Y=Z} m_1(x)\cdot m_2(y)$ which allows to implement events fusion. Furthermore, DSET calculates the belief about an event E, $Bel(E) = \sum_{X\sqsubseteq E} m(X)$ and its complement which is plausibility given by $Pls(E) = 1 - Bel(E) = 1 - \sum_{X\sqsubseteq E} m(X)$ The achieved data fusion with DSET must be associated with Fuzzy Logic to exclude imprecision using membership functions to filter imprecision in a rectifying manner. The design applies Fuzzy Logic to exclude imprecision, Dempster-Shafer Theory of Evidence to obtain fusion, and Frame-Based systems to apply medical constraints and personal preferences. Figure-3.14 depicts the interactions of the involved principles.

3.5.3 Mechanism-6: Knowledgeable representation and processing

Inference needs knowledge support developing additional artificial parameters as conclusions to Context-Aware Layer. The ontology must possess a structure:

- Upper Ontologies: manipulate concepts forming extensive conceptual hierarchies of relations among concepts and they are considered as “shallow”.

- Domain Ontologies: extend the Upper Ontologies to a particular domain providing constraints that form the framework for the Task Ontologies.
- Task Ontologies: oriented towards the inclusion of problem solving methods providing details within the space of Domain Ontologies with cardinality constraints, taxonomy of relations, and very specific constraints or axioms.
- Taxonomies: relations on specific domains of particular situations and problems

The system requires Thesauri services giving a registry of concepts facilitating communication and enriching the conceptual space with interfaces which is beyond the scope of this work. Formally, [3.2] ontology involves:

$$O := (C, H^C, R, H^R, I) \qquad \text{(Eq-3.43)}$$

where, C represents a set of concepts, H^C stands for the set of holding hierarchical relations among concepts, C used for the set relations among concepts, H^R represents a set of relations among the formed $H^R \sqsubseteq RxR$ relationships, and I stands for any instantiation of concepts in a specific domain at the particular task.

Knowledge is accessed through dictionary services as illustrated in Figure-3.15, to: transform developing new concepts while translate maps semantically equivalent entities or linguistic alternatives. Both operations achieve merging, integrating, mapping, and aligning ontologies with temporal tags creating knowledge versions.

3.5.4 Mechanism-9: Virtual context-awareness

Awareness is about the exhaustive availability of contextual information [3.9] involving the dimension d and x belongs to specific type of relations in d defining:

$$T = \{T_d | d \in DIM\} \qquad \text{(Eq-3.44)}$$

where, T refers to the set of types related with the dimensions in DIM which stands for the universe of dimensions.

The context C is defined as a finite relation

$$C = \{f(d,x) | d \in DIM \ \wedge x \colon T_d\} \qquad \text{(Eq-3.45)}$$

$$degree\ C = |dom\ C| \qquad \text{(Eq-3.46)}$$

with the degree of null C to be $degree\ C = 0$ denoting the empty relation.

Given a context $C = f(d_i, x_i) = \{(d_1, x_1), (d_2, x_2), \ldots, (d_n, x_n)\}$ then the function $\dim(\,)$ provides the involved dimensions and the function $tag(\,)$ provides the values associated with the dimensions in the context C with $C = \{f_i | f_i\} = (d_i, x_i)$ and correspondingly $\dim(C) = \{d_1, d_2, \ldots, d_i\}$ and $\text{tag}(C) = \{x_1, x_2, \ldots, x_i\}$. Thus, given a tuple $< d_k, x_k >$ of a context C then the dimension $\dim(C) = d_k$ and the corresponding values $\text{tag}(C) = x_k$ can be extracted. Hence, context can be approached as a construct of relations [3.9] which apply to manipulate its dynamic nature. In [3.9] contexts operators defined as follows:

- Override: the operator $\oplus$ receives two contexts and returns the first, the one of them overriding the second, i.e., $C_1 \oplus C_2 \colon C_1\; X\; C_2 \to C_1$
- Choice: the operator | receives a number of contexts and returns the selected one, i.e., $C_1 | C_2 | \ldots | C_k \colon C_1 X\; C_2\; X\; C_3\; X \ldots\ldots X\; C_k \to C_m$
- Projection: operator $\downarrow$ receives a context C and a set of dimensions $D \sqsubset DIM$ and filters only contexts with their dimensions in the given set, i.e., $C\; \downarrow D \colon C\; X\; D \to C.$
- Hiding: the operator $\uparrow$ receives a context C and applies on it a set of dimensions D to remove the primitive or atomic contexts whose dimensions are in D.
- Substitution: the operator $/$ receives a context C_1 and a context C_2 and returns context C_1 with replaced those sub-contexts of C_1 contained in C_2.
- Undirected range: the operator $\rightleftharpoons$ receives two contexts and returns a set of primitive or atomic contexts with an ordered set of tags.
- Directed range: the operator $\rightharpoonup$ receives two contexts and returns a set of the included sub-contexts between the given initial contexts.
- Difference: the operator $\ominus$ receives two contexts and returns the dimensions that do not belong to both of them.
- Join: the operator $\boxtimes$ unifies the received contexts
- Intersection: the operator $\boxdot$ returns the common dimensions
- Union: the operator $\boxplus$ returns the aggregation of the received contexts
- Box: is the operator that places in the same set the contexts of the same domain.

Mechanism-9 receives the virtual ACD structure enriched with knowledge from the used ontology extending ACD syntactically and semantically. The ontological relations determine the contextual operators in order to extend or create additional ACD structures creating the largest possible ACD dictating the contextual rules.

3.5.5 Mechanism-10: Decisive control

The topmost layer provides a virtual version of the physical situation enriched with relations and inferences as well as the necessary procedures to obtain the aimed goals. Mechanism-10 provides feedback instructions to the Physical Layer's entities considering the difference between the existing and the developing conceptual situations operating on concepts [3.9]. The input to Mechanism-10:
$V_{context} = Box(Virtual_{context})$ // set of primitive contexts
$P_{context} = Box(Physical_{context})$ // set of primitive contexts
$Primitives = V_{context} \ominus P_{context}$ // set of the difference of primitive contexts
$Input = Primitives \downarrow P_{context}$ // projection operation, interrelated primitives
// "*Input*" contains atomic contexts in Virtual Level and not present in Physical Level
$Contents \ = * \ Ontology(Input)$ // provide the constituents of each primitive
// "*Contents*" is a data structure containing attributes, properties, and functionality
$TowardsPhysicalLevel = * \ Contents[i]$ // provide input to each element in Physical Level
$Output = ControlOn(Contents)$ // applies controllability, observability, reachability, detectability and stabilizability tests
Mechanism-10 interchanges concepts and contexts considering that a context is a container superset of concepts with the uppermost layer corresponding to context.

3.6 Contextual Development

Context develops in steps starting from the identification of devices, applications, and people associated with a protocol for communication following the steps: (1) develop the ACD-structure for each entity, (2) compare attributes and properties of the ACDs to develop artificial parameters, (3) compare physical and artificial ACD to develop more parameters, and (4) the resulting set of data constitute the model's context.

The virtual counterpart of the physical model is more than a mapping due to the involved constraints applied on the contextual data and the supporting knowledge base achieving context-awareness and virtual correspondence requiring:

1) The elements of the developed context are examined through the filter consisted of the provided constraints to rectify the values of the artificial parameters.
2) The contextual structures are examined against the knowledge base applying the available constraints resulting into (a) additional artificial parameters or (b) adjust the attributes and values of the obtained artificial parameters.
3) The examination of the contextual structures returns the top most elements from each traversal of the knowledge hierarchy that meets the constraining requirements.
4) The resulting set of contextual attributes and the additionally provided artificial parameters from the knowledge base constitute the context-aware set.
5) The context-aware set is classified around the initially given data structures and the corresponding top most concepts from the knowledge base of the conceptual model.
6) Extended data structures and conceptual models develop the virtual model enriched by the ontologies and the constraints.

The virtual model contains: (i) the correspondence to the physical, (ii) an augmented structure, and (iii) conceptual instantiations with physical values. Two methods are required, setting structures of concepts from existing contexts, and forming contexts from existing concepts, traversing from context to concepts and visa-versa.

3.6.1 From context to concepts

The context can be expressed as a set of data structures

$$Context = C = \{c_{ij} | c \in \Theta \wedge i, j \in N\} \qquad \text{(Eq-3.47)}$$

and

$$c_{ij} = \{(c_{i1}, c_{i2}, \ldots., c_{ik}), (c_{21}, c_{22}, \ldots., c_{2k}), \ldots., \left(c_{j1}, c_{j2}, \ldots., c_{jm}\right)\} \qquad \text{(Eq-3.48)}$$

Then, a set of two computable functions can be defined referring to the constraints of the provided constraints and the associated knowledge, as it follows:

$$Constraints = f \colon N^k \to N \qquad \text{(Eq-3.49)}$$

$$Knowledge = g \colon N^k \to N \qquad \text{(Eq-3.50)}$$

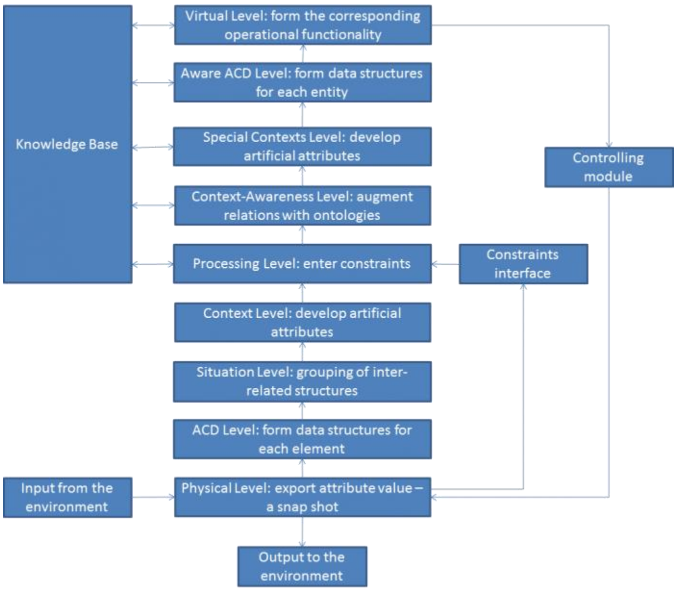

Knowledge Base
Virtual Level: form the corresponding operational functionality
Aware ACD Level: form data structures for each entity
Special Contexts Level: develop artificial attributes
Context-Awareness Level: augment relations with ontologies
Processing Level: enter constraints
Context Level: develop artificial attributes
Situation Level: grouping of inter-related structures
ACD Level: form data structures for each element
Physical Level: export attribute value – a snap shot
Input from the environment
Output to the environment
Controlling module
Constraints interface

Figure-3.16 The building blocks of the model’s architecture.

which provide a filter to the available context as

$$A(x) = (f \circ g)(x)c(x) \qquad \text{(Eq-3.51)}$$

The function composition of the constraints function f and the knowledge function g is applied on the context c resulting in the enriched version of the given context. Thus, a function is applied to either limit the functionality of c or augment its content.

3.6.2 From concepts to context

The decomposition of a conceptual abstraction to the contextual end involves the decomposition of the function representing concepts, $A(x)$. Since there is always available the function representing the actually physical set up by $f(x)$ and the knowledge level $g(x)$, then the context can be simply derived by

$$c(x) = \frac{A(x)}{(f \circ g)(x)} \qquad \text{(Eq-3.52)}$$

The reason for the establishment of a relationship from concepts to context is the availability of a test to ensure the proper transformation. At an instance of knowledge, the composition $(f \circ g)(x)$ can be estimated by the available conceptual information.

3.7 Functionality

The layered structure, Figure-3.16, provides at the lowest level the interfaces with inputs and outputs; it is supported by ontologies and two mechanisms, (i) input of constraints and (ii) output signals. The operation must be supported by memory and synchronizing modules.

Synopsis

Denotational Mathematics formally defines tools for the rigorous manipulation of concepts, systems, and processes. The determined tools apply on formally defined algebraic frameworks performing operations on concepts with Concept Algebra, carry out systemic functions with System Algebra, and executing processes with Process Algebra. The algebraic frameworks can describe and handle the carried functionality supported by appropriate architectural infrastructures. The supporting infrastructure must be capable of manipulating raw medical and environmental data to manage the developed context setting the formed system's components as context-aware. Context-awareness is used to achieve semantic interoperability among the participating devices and the coexisting software components. Thus, using Process Algebra achieves the manipulation of carried processes in order to obtain the functionality requirements of the infrastructure's components with System Algebra and to meet the desired medical conceptual setting administered with Concept Algebra. The employed infrastructure receives environmental home raw data and medical constraints from the supporting medical professionals which is transformed into information and then into knowledge. The obtained knowledge allows the development of a completely controllable virtual context upon which it takes place the system's decisions making. The drawn decisions and inferences through the infrastructure turn concepts into low level data to control the devices operating in the home environment supporting and satisfying the personal medical needs of the end-user. Decisions are made upon the developed virtual context using Control Theory principles to administer the operation and functionality of the devices operating in the home environment. Hence, the formally defined Denotational Mathematics algebraic tools applied on a matching architecture provides the capability to acquire the sensed medical and environmental parameters, develop the corresponding virtual medical context, make decisions based on the virtual context's parameters, and then, guide the dispersed ubiquitous computing devices to apply personalized medical treatment on the supporting users.

The Home UbiHealth model presents an adequate infrastructure to transform raw data into sound and complete concepts through which the model controls and interacts with the dispersed devices in the home environment supporting the personal medical needs of the end-user. In addition, the model must develop the virtual counterpart of the holding contextual situation and make adequate decisions in order to administer the health status of the supporting patient. The model must present the static and dynamic behaviors of the system's components in order to instruct accordingly and appropriately the co-operating devices by setting them as context-aware and behaving with the appropriate functionality to achieve the control of the end-user's health status.

CHAPTER 4. PROPOSED MODEL

The operation and the functional characteristics of the Home UbiHealth model rely on its static structures. The following steps provide the formal description of the model:

1. Layered Architectural Infrastructure: The layered conceptual structure of the model is formally described with Denotational Mathematics.
2. Closed Systemic Approach: The size and the complexity of the intrinsic characteristics of the model oblige to approach the design from the systems point of view. The model can be considered as a closed system as it considers the effects from all influencing parameters such as the remote medical offices, the regional hospitals and health centers, as well as the authorities and the related market of medical products and services.
3. Tools for self-Control: The necessary internal building components of the system of the Home UbiHealth model must be rigorously described with Denotational Mathematics which with their interactions provides the required property of autonomy. The required tools for the desired self-administration must be described with Denotational Mathematics which must be used by the embedded decisions making components.
4. Decisions Making: It is shown that the resulting structure is capable of receiving raw data from numerous sensing devices feeding the developed system which is capable of making adequate processing to make the proper decisions based on the use of the involved concepts.
5. From Data to Concepts: The system makes decisions based on the processing of concepts which are converted to data feeding the available actuating devices.
6. Constraints and Personal Preferences: The transparent operation of the system considers the medical constraints and the personal preferences of the supported individual with the aims to keep the health status the closest possible to the desired conditions.

The entire model is represented by:

$$\begin{aligned}\S(Home\_UbiHealth) \triangleq\ & Home\_UbiHealth\S.Architecture\mathbf{ST}\\ & \parallel Home\_UbiHealth\S.StaticBehavior\mathbf{PC}\\ & \parallel Home\_UbiHealth\S.DynamicBehavior\mathbf{PC}\end{aligned} \qquad \text{(Eq-4.1)}$$

In the following paragraphs it is given the architectural consideration of the Home UbiHealth model and the involved static behavior of its comprising components.

4.1 Layered Architectural Infrastructure

The description of the architectural components of the model is going to be defined using the Unified Data Models (UDM) constructs, also known as Component Logical Model (CLM) [4.1]. The identifiers of the employed UDM are as follows:

$$\begin{aligned} Home\_UbiHealth\S.Architecture\mathbf{ST} \triangleq \{ \quad & Layers\mathbf{ST} \\ & \| < Constraints\mathbf{ST} > < HoldingStatus\mathbf{ST} > \\ & \| \ SelfControl\mathbf{ST} \\ & \| \ KnowledgeBase\mathbf{ST} \\ & \| \ EventDriver\mathbf{ST} \ \} \end{aligned} \qquad \text{(Eq-4.2)}$$

The architecture of the Home UbiHealth model is a framework with four major components that carry out processes with their relations and interactions. The schematic presentation of the formed framework is presented in Figure-3.11 depicting the major layers of the structure supported by the associated software mechanisms. Each identifier in (Eq-4.2) refers to the models of the components constituting the architectural framework which are going to be further analyzed with the UDM contents. The framework of the layered structure is given by

$$Layer\mathbf{ST} \ \triangleq \S :: \{< {}^{TopLevel}_{i\mathbf{N=0}}R(Level[i\mathbf{N}]\mathbf{ST} >\} \qquad \text{(Eq-4.3)}$$

The introduced constraints are provided by properly structured interfaces that distinguished into

$$\begin{aligned} Constraints\mathbf{ST} \ \triangleq \S :: \{(& < MedicalConstraints\mathbf{ST} > \\ & \| < HealthStatus\mathbf{ST} > \\ & \| < PersonalPreferences\mathbf{ST} >) \\ & (< RuleBased\mathbf{ST} >)\} \end{aligned} \qquad \text{(Eq-4.4)}$$

The applied internal controlling tools on the processed data are enumerated and given by

$$SelfControl\mathbf{ST} \ \triangleq \S :: \{< {}^{Instrument}_{j\mathbf{N=0}}R(Tool(j\mathbf{N})\mathbf{ST}) >\} \qquad \text{(Eq-4.5)}$$

The distinct areas of the required knowledge is classified and provided by

$$KnowledgeBase\mathbf{ST} \triangleq \S :: \{< \underset{k\mathbf{N}=\mathbf{0}}{\overset{Taxonomies}{R}}(Knowledge(k\mathbf{N})\mathbf{ST} >\} \qquad \text{(Eq-4.6)}$$

The occurring events must be uniquely identified, classified, and processed accordingly

$$\begin{aligned} EventDriver\mathbf{ST} \triangleq \S :: \{ \; & < EventsRegistry\mathbf{ST} > \\ & \| < EventsInternal\mathbf{ST} > \\ & \| < EventsExternal\mathbf{ST} >\} \end{aligned} \qquad \text{(Eq-4.7)}$$

The individual's health status is part of the formed framework which is distinguished into the holding currently conditions, the desired conditions, and the reference conditions given by

$$\begin{aligned} HoldingStatus\mathbf{ST} \triangleq \S :: \{ \; & < CurrentStatus\mathbf{ST} > \\ & \| < DesiredStatus\mathbf{ST} > \\ & \| < ReferenceStatus\mathbf{ST} >\} \end{aligned} \qquad \text{(Eq-4.8)}$$

The interaction of the framework with the physical world is performed through the $EventsExternal\mathbf{ST}$ component of the $EventDriver\mathbf{ST}$ building block of the designed architecture. The $EventsExternal\mathbf{ST}$components is consisted of

$$\begin{aligned} & EventDriver\mathbf{ST}.EventsExternal\mathbf{ST} \triangleq \S :: \{ \\ & \qquad < RawDataPreProcessing\mathbf{ST} > \\ & \qquad \| < DataOutputProcessed\mathbf{ST} >\} \end{aligned} \qquad \text{(Eq-4.9)}$$

where,

$$\begin{aligned} RawDataPreProcessing\mathbf{ST} = \{ \; & < RawDataBuffering: \mathbf{ST} > \\ & \| < Fuzzification: \mathbf{ST} > \\ & \| < Fusion: \mathbf{ST} > \\ & \| < CommBankProtocols: \mathbf{ST} >\} \end{aligned} \qquad \text{(Eq-4.10a)}$$

and

$$\begin{aligned} & DataOutputProcessed\mathbf{ST} = (\\ & < ReceivedAtPosition\mathbf{ST}(< RecAddr: \mathbf{H}|0 \leq RecAddr\mathbf{H} \leq FFFF\mathbf{H} >) > \\ & < ProvidesAtPosition\mathbf{ST}(< ProvAddr: \mathbf{H}|0 \leq ProvAddr\mathbf{H} \leq FFFF\mathbf{H} >) >) \\ & < ReceivedType\mathbf{ST}(< RecType: \mathbf{N}|0 \leq RecType\mathbf{N} \leq m\mathbf{N} >) > \\ & < ProvidesValue\mathbf{ST}(< ProvValue: \mathbf{H}|0 \leq ProvValue\mathbf{H} \leq FFFF\mathbf{H} >) >) \end{aligned} \qquad \text{(Eq-4.10b)}$$

Then, the input data from the dispersed devices in the home environment are placed in discrete buffers positions $InBuffer\mathbf{ST},$ storing temporarily the input data $HoldValue\mathbf{ST}$ of magnitude or type $Magnitude\mathbf{ST}$ every $Sampling\mathbf{ST}$ intervals to time, and introducing the newly received data $OutBuffer\mathbf{ST}$ to the framework:

$RawDataBuffering\mathbf{ST} = ($

$< InBuffer\mathbf{ST}(< InAddr: \mathbf{H}|0 \leq InAddr\mathbf{H} \leq FFFF\mathbf{H} >) >$

$< HoldValue\mathbf{ST}(< ReadValue: \mathbf{H}|0 \leq ReadValue\mathbf{H} \leq FFFF\mathbf{H} >) >$

$< Magnitude\mathbf{ST}(< Magnitude: \mathbf{N}|0 \leq Magnitude\mathbf{N} \leq m\mathbf{N} >) >$

$< Sampling\mathbf{ST}(< Rate: \mathbf{N}|t_1\mathbf{N} \leq Rate\mathbf{N} \leq t_2\mathbf{N} >) >$

$< OutBuffer\mathbf{ST}(< OutAddr: \mathbf{H}|0 \leq OutAddr\mathbf{H} \leq FFFF\mathbf{H} >) >)$ (Eq-4.11)

The acquired data feed the component at $FuzzyInPort\mathbf{ST}$ that applies fuzzy filtering with $FuzzyScale\mathbf{ST}$ on the received data in order to exclude imprecisions caused by the involved noise or interferences providing at $FuzzyOutPort\mathbf{ST}$ the obtained value of $FuzzyOutput\mathbf{ST}$ for further processing as follows:

$Fuzzification\mathbf{ST} = ($

$< FuzzyInPort\mathbf{ST}(< FuzInAddr: \mathbf{H}|0 \leq FuzInAddr\mathbf{H} \leq FFFF\mathbf{H} >) >$

$< FuzzyScale\mathbf{ST}(< FuzScale: \mathbf{H}|0 \leq FuzScale\mathbf{H} \leq FFFF\mathbf{H} >) >$

$< FuzzyOutput\mathbf{ST}(< FuzValue: \mathbf{N}|0 \leq FuzValue\mathbf{N} \leq m\mathbf{N} >) >$

$< FuzzyOutput\mathbf{ST}(< Rate: \mathbf{N}|t_1\mathbf{N} \leq Rate\mathbf{N} \leq t_2\mathbf{N} >) >$

$< FuzzyOutput\mathbf{ST}(< FuzOutAddr: \mathbf{H}|0 \leq FuzOutAddr\mathbf{H} \leq FFFF\mathbf{H} >) >)$ (Eq-4.12)

The fuzzy values of the acquired raw data must be fed into the next component to perform the operation of fusion with $Fusion\mathbf{ST}$ since there can be more than one source of data providing the same type of signal. Thus, the fusion parameters must be available including the $Value\mathbf{N}, Weight\mathbf{N},$ and the $Position\mathbf{N}$ in order to obtain the dominating value for each type of read data source. The received values are placed at $FusionInPort\mathbf{ST}$ and the fused value $FusedValue\mathbf{ST}$ is received at $FusionOut\mathbf{ST}$ for further processing:

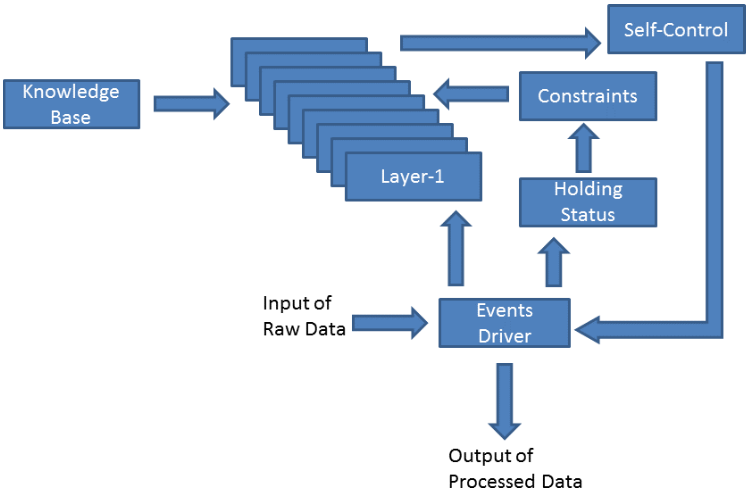


Figure-4.1 The architectural building blocks of the model.

$$
\begin{aligned}
&Fusion\mathbf{ST} = (\\
&< FusionInPort\mathbf{ST}(< FusInAddr: \mathbf{H}|0 \leq FusInAddr\mathbf{H} \leq FFFF\mathbf{H} >) >\\
&< FuzzyParameters\mathbf{ST}(\\
&\qquad\qquad < Weight: \mathbf{H}|0 \leq Weight\mathbf{H} \leq 1\mathbf{N} >\\
&\qquad\qquad < Value: \mathbf{H}|0 \leq Value\mathbf{H} \leq FF\mathbf{H} >\\
&\qquad\qquad < Position: \mathbf{H}|0 \leq Position\mathbf{H} \leq FF\mathbf{H} >)\\
&< FusedValue\mathbf{ST}(\\
&\qquad\qquad < Val: \mathbf{H}|0 \leq Val\mathbf{H} \leq FF\mathbf{H} >\\
&\qquad\qquad < Magn: \mathbf{H}|0 \leq Magn\mathbf{H} \leq FF\mathbf{H} >)\\
&< FusionOutPort\mathbf{ST}(< FusOutAddr: \mathbf{H}|0 \leq FusOutAddr\mathbf{H} \leq FFFF\mathbf{H} >) >) \quad \text{(Eq-4.13)}
\end{aligned}
$$

The $RawDataPreProcessing\mathbf{ST}$ component is supported by a Bank of Communication Protocols $CommBankProtocols\mathbf{ST}$ that dictates the administration, by the $Protocol\mathbf{ST}$ component, of each input source of data. For each input type the identification, $PositionId\mathbf{H}$, the associated rules, $Rule\mathbf{H}$, and the physical magnitude, $PhysMagn\mathbf{H}$, are specified in order to facilitate the administrative handling of each known source of data:

$$
\begin{aligned}
&CommBankProtoco\mathbf{ST} = (< CommBankProtoco\mathbf{ST}(\\
&\quad < PositionId: \mathbf{H}|0 \leq PositionId\mathbf{H} \leq FF\mathbf{H} >\\
&\quad < Rules: \mathbf{H}|0 \leq Rules\mathbf{H} \leq FF\mathbf{H} >\\
&\quad < PhysMagn: \mathbf{H}|0 \leq PhysMagn\mathbf{H} \leq FF\mathbf{H} >) \qquad \text{(Eq-4.14)}
\end{aligned}
$$

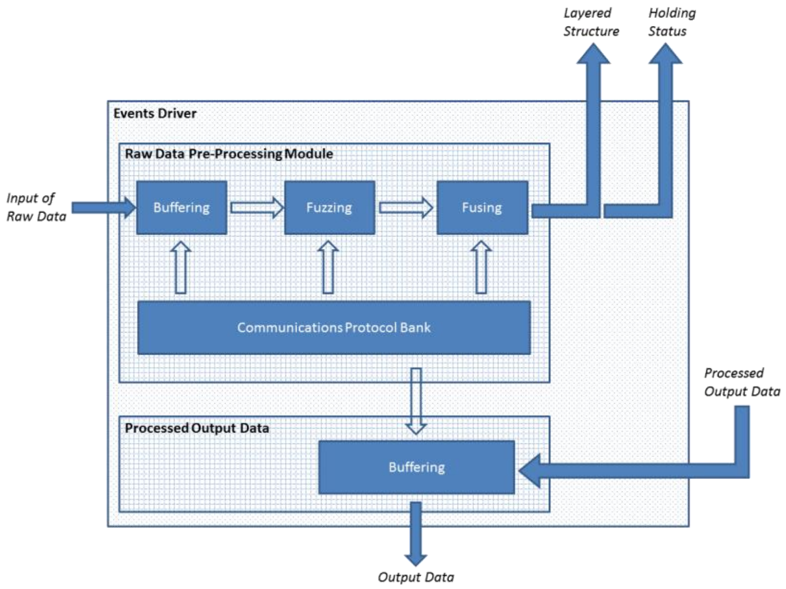


Figure-4.2 The architectural building blocks of the $EventsExternal\mathbf{ST}$ component.

The architectural building blocks of the model are presented in Figure-4.1 depicting the components involved into (Eq-4.2).

The $EventDriver\mathbf{ST}.EventsExternal\mathbf{ST}$ performs two operations. Firstly, it performs the pre-processing operation of the received raw data facilitated by the $RawDataPreProcessing\mathbf{ST}$ component. Secondly, the $DataOutputProcessed\mathbf{ST}$ component provides the processed data at the physical world's interfaces. The diagram in Figure-4.2 presents the involved architectural building components described by (Eq-4.9 through 4.14).

4.1.1 Design of the stack of Layers

The description of (Eq-4.3) provides a condensed expression for the layered infrastructure that provides a sequential mode of processing on the prepared raw data. Iterating through each loop of (Eq-4.3), each layer of the infrastructure can be visited and examined. The first layer in the infrastructure refers to the counterpart of the existing physical set-up of devices and software entities which are presented as it follows:

$Level[0]\mathbf{ST} = (< PhysicalLevel\mathbf{ST}($

$< Interface[0]\mathbf{ST}($

$< InputInterface\mathbf{ST}($

$< InMemPosit: \mathbf{H} | 0 \leq InMemPosit\mathbf{H} \leq FF\mathbf{H} >$

$< InMemVal{:}\,\mathbf{H}|0 \leq InMemVal\mathbf{H} \leq FF\mathbf{H} >$

$< InResource{:}\,\mathbf{H}|0 \leq InResource\mathbf{H} \leq FF\mathbf{H} >) >$

$< OutputInterface\mathbf{ST}($

$< OutMemPosit{:}\,\mathbf{H}|0 \leq OutMemPosit\mathbf{H} \leq FF\mathbf{H} >$

$< OutMemVal{:}\,\mathbf{H}|0 \leq OutMemVal\mathbf{H} \leq FF\mathbf{H} >$

$< OutResource{:}\,\mathbf{H}|0 \leq OutResource\mathbf{H} \leq FF\mathbf{H} >) >$

$) >$

$< ProcessingSpace[0]\mathbf{ST}($

$< MemPointer{:}\,\mathbf{H}|0 \leq MemPointer\mathbf{H} \leq FF\mathbf{H} >) >$

$) >$ (Eq-4.15a)

In the same fashion, the internal structures of each of the rest of the infrastructure's layers are defined. It follows the definition of the architectural form of the layers in (Eq-4.15b) through (Eq-4.15i).

$$Level[1]\mathbf{ST} = (< ACDLevel\mathbf{ST}(\; < Interface[1]\mathbf{ST} > \; < ProcessingSpace[1]\mathbf{ST} >) > \tag{Eq-4.15b}$$

$$Level[2]\mathbf{ST} = (< SituationLevel\mathbf{ST}(\; < Interface[2]\mathbf{ST} > \; < ProcessingSpace[2]\mathbf{ST} >) > \tag{Eq-4.15c}$$

$$Level[3]\mathbf{ST} = (< ContextLevel\mathbf{ST}(\; < Interface[3]\mathbf{ST} > \; < ProcessingSpace[3]\mathbf{ST} >) > \tag{Eq-4.15d}$$

$$Level[4]\mathbf{ST} = (< ProcessingLevel\mathbf{ST}(\; < Interface[4]\mathbf{ST} > \; < ProcessingSpace[4]\mathbf{ST} >) > \tag{Eq-4.15e}$$

$$Level[5]\mathbf{ST} = (< AwarenessLevel\mathbf{ST}(\; < Interface[5]\mathbf{ST} > \; < ProcessingSpace[5]\mathbf{ST} >) > \tag{Eq-4.15f}$$

$$Level[6]\mathbf{ST} = (< SpecialLevel\mathbf{ST}(\; < Interface[6]\mathbf{ST} > \; < ProcessingSpace[6]\mathbf{ST} >) > \tag{Eq-4.15g}$$

$$Level[7]\mathbf{ST} = (< AwareACDLevel\mathbf{ST}(\; < Interface[7]\mathbf{ST} >$$

$$< ProcessingSpace[7]\mathbf{ST} >) > \qquad \text{(Eq-4.15h)}$$

$$Level[8]\mathbf{ST} = (< VirtualLevel\mathbf{ST}($$
$$< Interface[8]\mathbf{ST} >$$
$$< ProcessingSpace[8]\mathbf{ST} >) > \qquad \text{(Eq-4.15i)}$$

Figure-4.3 depicts the building blocks of a typical layer and the layers of the stack.

4.1.2 Implementing the applied Constraints

The inserted and applied constraints have three aspects, first, the holding conditions of the individual's health status that can request certain actions to bring the system to an equilibrium state, second, the personal preferences of the supported individual that aim to comfort living at home, and third, the medical requirements as a result of the diagnosed diseases for the supported individual. The framework is connected to properly adapted interfaces to receive the raw data from all these three sources that constitute the applied constraints on the decisions made to eventually support the individual's health status at home. Analyzing the constituting components of (Eq-4.4), the applied constraints must present the following input, processing, and output facilities. The medical constraints consists of

$$MedicalConstraints\mathbf{ST} = (< Diagnostics\mathbf{ST} >< Diagnoses\mathbf{ST} >$$
$$< Therapies\mathbf{ST} > < Drugs\mathbf{ST} >< CurePlan\mathbf{ST} >) \text{(Eq-4.16a)}$$

The constraints applied by the current status of the individual's health is provided by the following relationship that provides three aspects, first, the desired status set by the medical professionals, second, the reference status referring to the normal values of the health status set by the medical professionals, and third, the holding conditions of the individual's health status provided by the evidences from the ubiquitous computing environment,

$$HealthStatus\mathbf{ST} = (< Desired\mathbf{ST} >< Reference\mathbf{ST} >< Holding\mathbf{ST} >) \text{ (Eq-4.16b)}$$

The last set of parameters that apply constraints is related with the personal preferences of the individual. This set refers to the energy administration, the holding conditions at home, the list of the activities of daily living (ADL), the entertainment activity, and the social security characteristics of the individual,

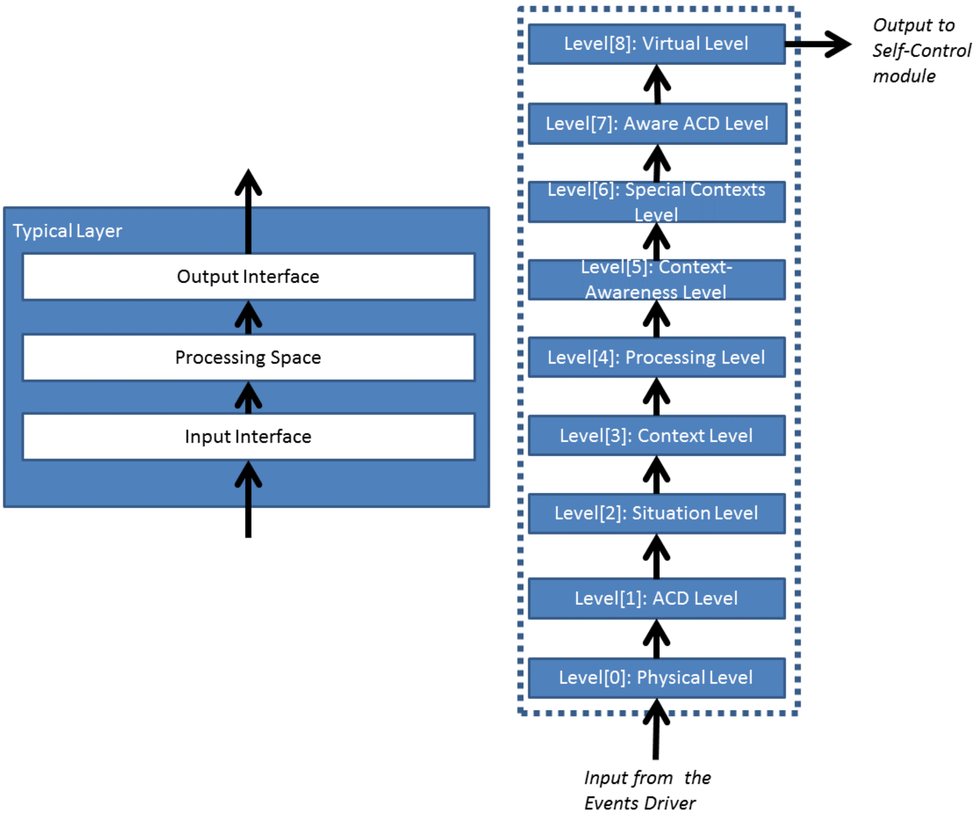


Figure-4.3 The architectural building blocks of a typical layer.

$$PersonalPreferences\mathbf{ST} = (< Energy\mathbf{ST} >< HomeConditions\mathbf{ST} > \\ < ADL\mathbf{ST} > < Entertainment\mathbf{ST} >< SocialSecurity\mathbf{ST} >) \text{(Eq-4.16c)}$$

Each of the employed components is further analyzed into three major blocks: the input buffer, the processing space, and the output buffer. For instance, the $Diagnostics\mathbf{ST}$ component is directly proportional to the typical component that is consisted:

$$< Diagnostics\mathbf{ST} = (< InputDiagnostics\mathbf{ST}($$

$$< InDiagnMemPosit\colon \mathbf{H}|0 \leq InDiagnMemPosit\mathbf{H} \leq FF\mathbf{H} >$$

$$< InDiagnMemVal\colon \mathbf{H}|0 \leq InDiagnMemVal\mathbf{H} \leq FF\mathbf{H} >$$

$$< InDiagnResource\colon \mathbf{H}|0 \leq InDiagnResource\mathbf{H} \leq FF\mathbf{H} >) >$$

$$< ProcessingSpaceDiagnostic[0]\mathbf{ST}($$

$$< DiagnMemPointer\colon \mathbf{H}|0 \leq DiagnMemPointer\mathbf{H} \leq FF\mathbf{H} >) >$$

$$< OutputDiagnostics\mathbf{ST}($$

$$< OutDiagnMemPosit\colon \mathbf{H}|0 \leq OutDiagnMemPosit\mathbf{H} \leq FF\mathbf{H} >$$

$$< OutDiagnMemVal\colon \mathbf{H}|0 \leq OutDiagnMemVal\mathbf{H} \leq FF\mathbf{H} >$$

$$< OutDiagnResource\colon \mathbf{H}|0 \leq OutDiagnResource\mathbf{H} \leq FF\mathbf{H} >) >$$

$$) > \qquad \text{(Eq-4.17)}$$

The output buffers feed a rules-based module that introduces the set of constraints which is going to be applied onto the data of the layered framework. The rules-based module is consisted of three components including: first, the rules conditions component, second, the processing memory space module accessed by the inference engine, and third, the inference component,

$$\begin{array}{l}
RulesBased\mathbf{ST} = (\\
\quad < InRulesBased\mathbf{ST}(\\
\qquad < InRulesBasedMemPosit{:}\,\mathbf{H}|0 \leq InRulesBasedMemPosit\mathbf{H} \leq FF\mathbf{H} > \\
\qquad < InRulesBasedMemVal{:}\,\mathbf{H}|0 \leq InRulesBasedMemVal\mathbf{H} \leq FF\mathbf{H} >) > \\
\quad < OutRulesBased\mathbf{ST}(\\
\qquad < OutRulesBasedMemPosit{:}\,\mathbf{H}|0 \leq OutRulesBasedMemPosit\mathbf{H} \leq FF\mathbf{H} > \\
\qquad < OutRulesBasedMemVal{:}\,\mathbf{H}|0 \leq OutRulesBasedMemVal\mathbf{H} \leq \\
FF\mathbf{H} >) > \\
\quad < InferenceSpace[0]\mathbf{ST}(\\
\qquad < MRulesBasedemPointer{:}\,\mathbf{H}|0 \leq RulesBasedMemPointer\mathbf{H} \leq FF\mathbf{H} >) > \\
) >
\end{array} \qquad \text{(Eq-4.18)}$$

In Figure-4.4 the component of (Eq-4.4) are depicted where the applied medical, the sensed health status, and the personal preferences constitute the in-parallel acting components that feed a rule-based system.

4.1.3 Regulating Component

The model has the capability to adjust the values of its parameters in order to satisfy its purpose and attain its goals related to the individual's health status. The discrete values of the model's output are directed to the dispersed devices affecting with their operation the individual's context. The model adjusts its operation using the acquired raw data which is processed with the held knowledge to provide the proper instructions to the operating physical devices. The input to the regulating or self-controlling component is received from the virtually developed aware context which provides the required values to the actuating physical devices. The regulating component is equipped with a set of tools borrowed from Control Theory focusing on Controllability, Observability, Stabilizability, Maintainability, Stability,

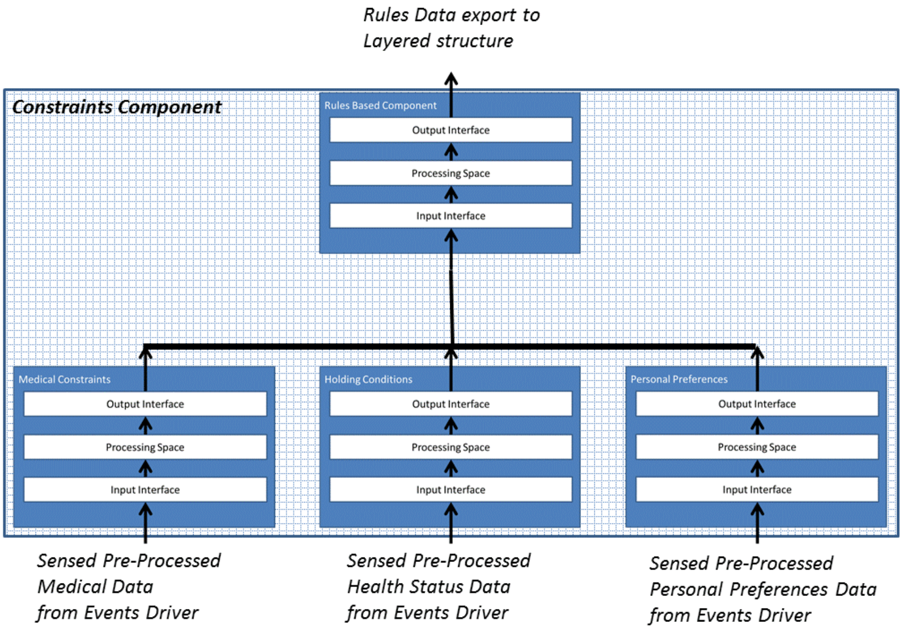


Figure-4.4 The architectural building blocks of the constraints component.

Detectability, and Reachability. Expanding (Eq-4.5), it reveals the associated tools used for the adjustment of the physical devices operational behavior as it

$$
\begin{aligned}
&Self\,Control\mathbf{ST} \triangleq \S :: (\\
&\quad < InDataFromVirtualContext\mathbf{ST} > \\
&\quad < Controller\mathbf{ST} > \\
&\quad < ControlingTools\mathbf{ST}(\\
&\quad (< ToolControllability\mathbf{ST} >< ToolObservability\mathbf{ST} >) \\
&\| \ (\ < ToolStabilizability\mathbf{ST} >< ToolMaintainability\mathbf{ST} >< \\
&\qquad ToolStability\mathbf{ST} >< ToolDetectability\mathbf{ST} >< ToolReacability\mathbf{ST} >) >) \\
&< OptimalDecision\mathbf{ST} >) \qquad \text{(Eq-4.19)}
\end{aligned}
$$

The corresponding block diagram is depicted in Figure-4.5 presenting the input and output components along with the employed structures of the applied tools. The structure of the self-controlling component is equipped with a decision making component that provides the data representing the taken decisions to the Events Driver component to reach the interfaces of physical devices of the model.

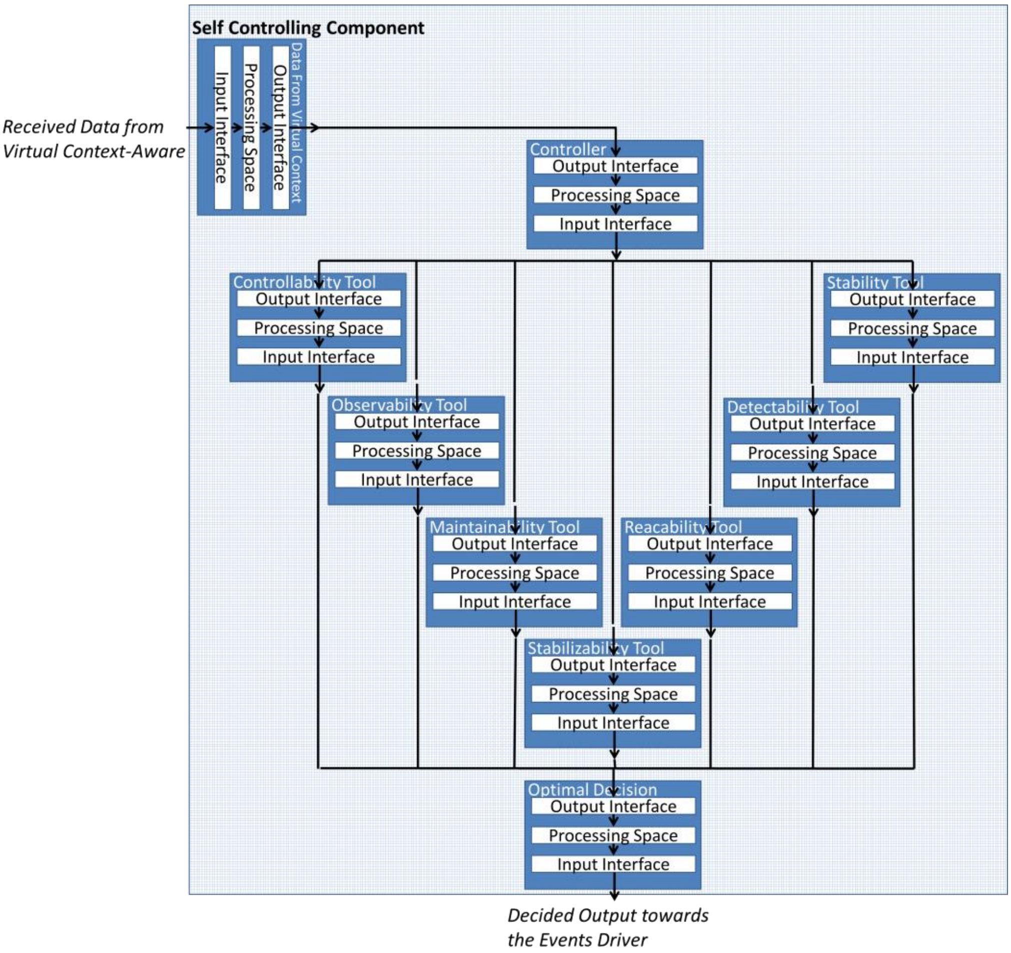


Figure-4.5 The architectural building blocks of the self-controlling component.

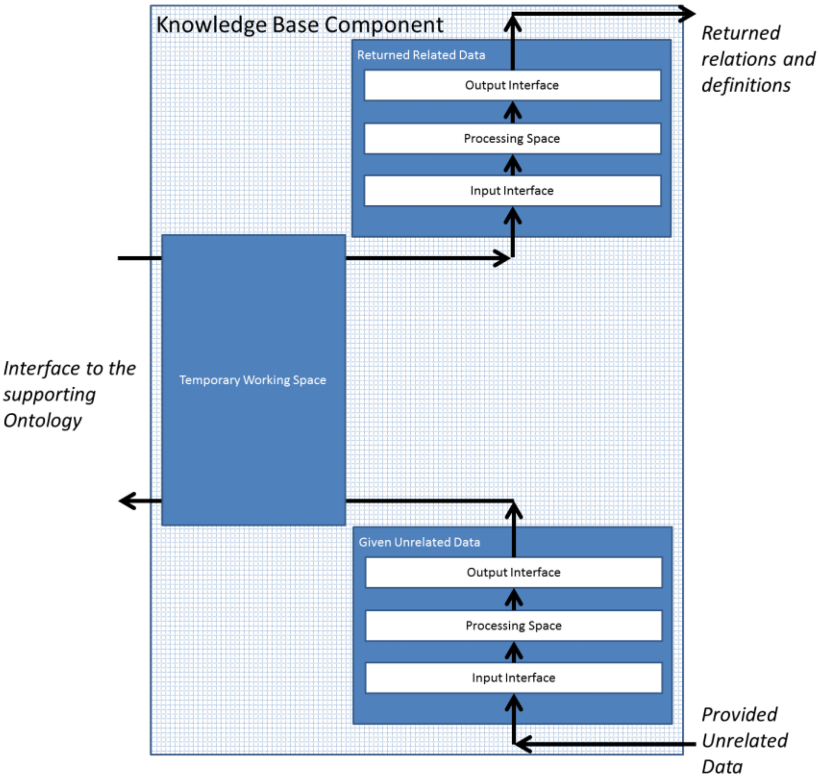


Figure-4.6 The architectural blocks of the knowledge base interfacing component.

4.1.4 Knowledge Base Component

The model's framework must be supported by a knowledge base in order to achieve the integration of raw data into information and then the transformation of information to wisdom. The contribution of the required knowledge base aims at two necessary constituents: first, relations among raw data without direct but implied relationships, and second, the definition of terms, composite elements, and concepts along with their established relations,

$$\begin{aligned} KnowledgeBase\mathbf{ST} \triangleq \S :: \{< {}^{Taxonomies}_{k\mathbf{N}=\mathbf{0}}R(Knowledge(k\mathbf{N})\mathbf{ST} >\} :: (& \\ < GivenUnrelatedData\mathbf{ST} >& \\ < TemporaryWorkingSpace\mathbf{ST} >& \\ < ReturnedRelatedData\mathbf{ST} >)& \qquad \text{(Eq-4.20)} \end{aligned}$$

The block diagram describing the infrastructure of the knowledge base component is given in Figure-4.6 presenting the constituting building blocks.

4.1.5 Data Transformations

The set of raw data S_1 is obtained from the physical layer constituting the first level L_1 which contains the cooperating devices in the ubiquitous computing home environment. Processing the elements of S_1, the set S_2 is obtained and applying successive processing on the obtained set the following chain of relations is defined forming a sequence of functions:

obtaining the set of ACD structures,

$$f_1 : S_1 \rightarrow S_2 \qquad \text{(Eq-8.21a)}$$

forming the set of Active Space (AS) structures,

$$f_2 : S_2 \rightarrow S_3 \qquad \text{(Eq-8.21b)}$$

forming the content of the ubiquitous computing situations,

$$f_3 : S_3 \rightarrow S_4 \qquad \text{(Eq-8.21c)}$$

developing the corresponding context,

$$f_4 : S_4 \rightarrow S_5 \qquad \text{(Eq-8.21d)}$$

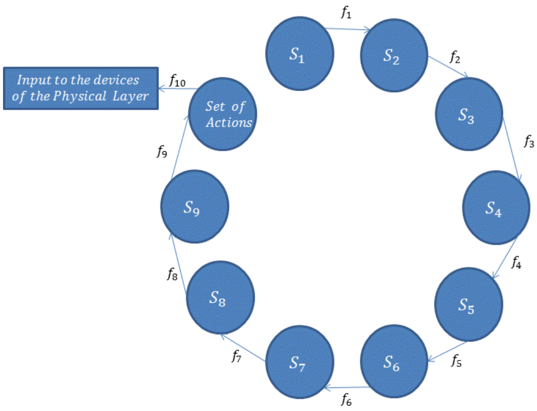


Figure-4.7 The sequence of data transformations.

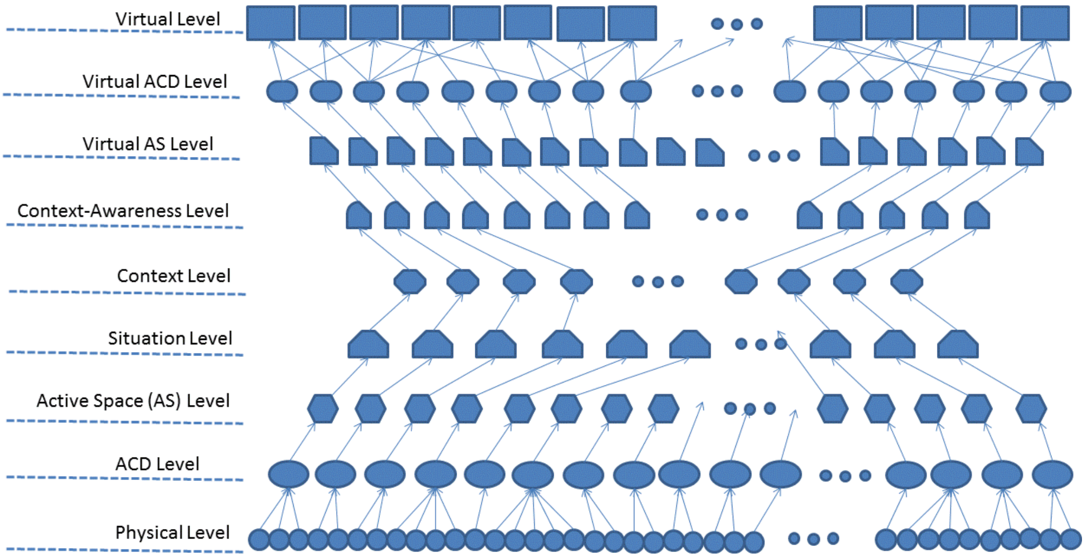


Figure-4.8 The data transformation performed by the layers of the system.

enriching the contextual context with knowledgeable relations,

$$f_5 \colon S_5 \to S_6 \tag{Eq-8.21e}$$

developing the corresponding aware AS structures,

$$f_6 \colon S_6 \to S_7 \tag{Eq-8.21f}$$

forming the corresponding aware ACD structures and

$$f_7 \colon S_7 \to S_8 \tag{Eq-8.21g}$$

developing the virtual aware context.

$$f_8 \colon S_8 \to S_9 \tag{Eq-8.21h}$$

On the obtained virtual context-aware set S_9, there are applied control tools to make decision on how to instruct the physical devices to act at the physical layer. In Figure-4.7 it is depicted the transformation of the given physical set of data until it receives back instructions for the operating devices functioning at the physical layer supporting the individual's health at the home ubiquitous computing environment.

4.2 Closed Systemic approach

The data from the dispersed physical devices in the ubiquitous computing home environment constitute the physical level of the model. The raw data from the devices are collected in the physical layer and then they are transformed successively to the next layers up to the formation of the virtual corresponding layers. The virtual layer of the model corresponds to the physical layer enriched by artificial parameters as a result to the holding interrelationships. The virtual layer allows the manipulation of complete information and concepts instead of using detailed data. Figure-4.8 depicts the transformations as the data pass through each level of the model's internal layers. A process is applied at each level of the model augmenting the contained data with the holding relations among the parameters and with the knowledge provided by the associated knowledge base. The applied processes consider the set of medical constraints and the personal preferences of the supported individual. The layers, the supporting knowledge base and the constraints constitute a system since the content of the virtual layer provides guidance to the devices of the physical layer.

The availability of the virtual layer provides the required data to examine the effects on the individual's health status. The effects on the individual's health are examined performing a set of tests applied on the formed system. It follows the application of controlling tests and the actions taken to support the individual's health. The model is examined, tested, and controlled applying manipulations on the following set of simultaneous equations that represent the system:

$$\frac{dx}{dt} = \bar{A}x + \bar{B}u$$
$$y = \bar{C}x + \bar{D}u \qquad \text{(Eq-4.22)}$$

In (Eq-4.22) there is large number of parameters involved that make the solution of the set of simultaneous equations impractical due to the formation of large matrices as in $\bar{A}, \bar{B}, \bar{C}, \bar{D}$ requiring computing intensity and considerable time to perform the necessary inversions. Hence, alternative approaches must be considered leading to the use of tools from Control Theory since the contents of the $\bar{A}, \bar{B}, \bar{C}, \bar{D}$ matrices is already available at the virtual layers of the model.

4.2.1 Testing the System status

The self-control component of the model contains the required tools to make the appropriate optimal decisions. The decisions are related with the issuance of specific instructions which are going to be directed at specific devices. The instructions directed to the devices at the physical layer are chosen be such that the status of the system is at one of the desired states. The desired system states are directly proportional to the health status of the individual considering the medical constraints; the holding conditions are home, as well as, the personal preferences. In order to characterize the status of the system, the testing tools borrowed from Control Theory must be applied with the characteristics of Controllability test to be given

$ToolControllability\mathbf{ST} = \quad ($

$< InputInterface\mathbf{ST}($

$< d_{Afrom}: \mathbf{H} | 0 \leq d_{Afrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Ato}: \mathbf{H} | 0 \leq d_{Ato}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Bfrom}: \mathbf{H} | 0 \leq d_{Bfrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Bto}: \mathbf{H} | 0 \leq d_{Bto}\mathbf{H} \leq FF\mathbf{H} >$

$< InPointerMatrixA: \mathbf{H} | [d_{Afrom}][d_{Ato}] >$

$< InPointerMatrixB: \mathbf{H} | [d_{Bfrom}][d_{Bto}] >$

$< OutputInterface\mathbf{ST}($

$< OutRankMatrixP_{Contr}: \mathbf{H} | 0 \leq OutRankMatrixP_{Contr} \leq FF\mathbf{H} >$

$< ProcessingSpace\mathbf{ST}($

$< d_{Cfrom}: \mathbf{H} | 0 \leq d_{Cfrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Cto}: \mathbf{H} | 0 \leq d_{Cto}\mathbf{H} \leq FF\mathbf{H} >$

$< MatrixP_{Contr}: \mathbf{H} \big| \big[d_{Cfrom}\big][d_{Cto}] > \big) > \ )$ (Eq-4.23a)

Similarly, the tool for Observability is given by:

$ToolObservability\mathbf{ST} = \quad ($

$< InputInterface\mathbf{ST}($

$< d_{Afrom}: \mathbf{H} | 0 \leq d_{Afrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Ato}\colon \mathbf{H}|0 \leq d_{Ato}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Cfrom}\colon \mathbf{H}|0 \leq d_{Cfrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Cto}\colon \mathbf{H}|0 \leq d_{Cto}\mathbf{H} \leq FF\mathbf{H} >$

$< InPointerMatrixA\colon \mathbf{H}|[d_{Afrom}][d_{Ato}] >$

$< InPointerMatrixC\colon \mathbf{H}|[d_{Cfrom}][d_{Cto}] >$

$< OutputInterface\mathbf{ST}($

$< OutRankMatrixP_{Observ}\colon \mathbf{H}|0 \leq OutRankMatrixP_{Observ} \leq FF\mathbf{H} >$

$< ProcessingSpace\mathbf{ST}($

$< d_{Cfrom}\colon \mathbf{H}|0 \leq d_{Cfrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Cto}\colon \mathbf{H}|0 \leq d_{Cto}\mathbf{H} \leq FF\mathbf{H} >$

$< MatrixP_{Observ}\colon \mathbf{H}\big|\big[d_{Cfrom}\big][d_{Cto}] >\big) >)$ (Eq-4.23b)

In the same fashion, the tool for Stabilizability is provided when $C = [A - \lambda I \quad B]$full row rank has given

$ToolStabilizability\mathbf{ST} = \quad ($

$< InputInterface\mathbf{ST}($

$< d_{Afrom}\colon \mathbf{H}|0 \leq d_{Afrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Ato}\colon \mathbf{H}|0 \leq d_{Ato}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Bfrom}\colon \mathbf{H}|0 \leq d_{Bfrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Bto}\colon \mathbf{H}|0 \leq d_{Bto}\mathbf{H} \leq FF\mathbf{H} >$

$< \lambda\colon \mathbf{H}|0 \leq \lambda\mathbf{H} \leq FF\mathbf{H} >$

$< InPointerMatrixA\colon \mathbf{H}|[d_{Afrom}][d_{Ato}] >$

$< InPointerMatrixC\colon \mathbf{H}|[d_{Bfrom}][d_{Bto}] >$

$< OutputInterface\mathbf{ST}($

$< OutRankMatrixP_{Stabil}\colon \mathbf{H}|0 \leq OutRankMatrixP_{Stabil} \leq FF\mathbf{H} >$

$< ProcessingSpace\mathbf{ST}($

$< d_{Cfrom}\colon \mathbf{H}|0 \leq d_{Cfrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Cto}\colon \mathbf{H}|0 \leq d_{Cto}\mathbf{H} \leq FF\mathbf{H} >$

$< MatrixP_{Stabil}\colon \mathbf{H}\big|\big[d_{Cfrom}\big][d_{Cto}] >\big) >)$ (Eq-4.23c)

The Maintainability tool refers to the capability of the model to correct faults, improve performance, change its attributes, or adapt to the changes of the environment in order to obtain a stable model and it is provided

$$ToolMaintainability\mathbf{ST} = \quad ($$

$$< InputInterface\mathbf{ST}($$

$$< d_{Afrom}: \mathbf{H}|0 \leq d_{Afrom}\mathbf{H} \leq FF\mathbf{H} >$$

$$< d_{Ato}: \mathbf{H}|0 \leq d_{Ato}\mathbf{H} \leq FF\mathbf{H} >$$

$$< d_{Bfrom}: \mathbf{H}|0 \leq d_{Bfrom}\mathbf{H} \leq FF\mathbf{H} >$$

$$< d_{Bto}: \mathbf{H}|0 \leq d_{Bto}\mathbf{H} \leq FF\mathbf{H} >$$

$$< InPointerMatrixA: \mathbf{H}|[d_{Afrom}][d_{Ato}] >$$

$$< InPointerMatrixC: \mathbf{H}|[d_{Bfrom}][d_{Bto}] >$$

$$< OutputInterface\mathbf{ST}($$

$$< OutRankMatrixP_{Maint}: \mathbf{H}|0 \leq OutRankMatrixP_{Maint} \leq FF\mathbf{H} >$$

$$< ProcessingSpace\mathbf{ST}($$

$$< d_{Cfrom}: \mathbf{H}|0 \leq d_{Cfrom}\mathbf{H} \leq FF\mathbf{H} >$$

$$< d_{Cto}: \mathbf{H}|0 \leq d_{Cto}\mathbf{H} \leq FF\mathbf{H} >$$

$$< MatrixP_{Maint}: \mathbf{H}\big|\big[d_{Cfrom}\big][d_{Cto}] >\big) >) \qquad \text{(Eq-4.23c)}$$

The Stability tool takes advantage of the fact that the eigenvalues of matrix $\bar{A}$ have negative real part and it is provided

$$ToolStability\mathbf{ST} = \quad ($$

$$< InputInterface\mathbf{ST}($$

$$< d_{Afrom}: \mathbf{H}|0 \leq d_{Afrom}\mathbf{H} \leq FF\mathbf{H} >$$

$$< d_{Ato}: \mathbf{H}|0 \leq d_{Ato}\mathbf{H} \leq FF\mathbf{H} >$$

$$< \lambda: \mathbf{N}|\lambda\mathbf{N} \leq 0 >$$

$$< InPointerMatrixA: \mathbf{H}|[d_{Afrom}][d_{Ato}] >$$

$$< OutputInterface\mathbf{ST}($$

$$< EigenValueRealPart\lambda: \mathbf{H}|0 \leq EigenValueRealPart\lambda \leq FF\mathbf{H} >$$

$$< ProcessingSpace\mathbf{ST}($$

$$< d_{Afrom}: \mathbf{H}|0 \leq d_{Afrom}\mathbf{H} \leq FF\mathbf{H} >$$

$$< d_{Ato}: \mathbf{H}|0 \leq d_{Ato}\mathbf{H} \leq FF\mathbf{H} >$$

$$< MatrixA: \mathbf{H}\big|\big[d_{Afrom}\big][d_{Ato}] >\big) >) \qquad \text{(Eq-4.23e)}$$

The tool for testing the Detectability property is using the Hautus test which refers to $[\begin{matrix} A - \lambda I \\ C \end{matrix}]$ has full column rank, it is given

$ToolDetectability\mathbf{ST} = \quad ($

$< InputInterface\mathbf{ST}($

$< d_{Afrom}: \mathbf{H}|0 \leq d_{Afrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Ato}: \mathbf{H}|0 \leq d_{Ato}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Cfrom}: \mathbf{H}|0 \leq d_{Cfrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Cto}: \mathbf{H}|0 \leq d_{Cto}\mathbf{H} \leq FF\mathbf{H} >$

$< \lambda: \mathbf{H}|0 \leq \lambda\mathbf{H} \leq FF\mathbf{H} >$

$< InPointerMatrixA: \mathbf{H}|[d_{Afrom}][d_{Ato}] >$

$< InPointerMatrixC: \mathbf{H}|[d_{Cfrom}][d_{Cto}] >$

$< OutputInterface\mathbf{ST}($

$< OutRankMatrixP_{Detect}: \mathbf{H}|0 \leq OutRankMatrixP_{Detect} \leq FF\mathbf{H} >$

$< ProcessingSpace\mathbf{ST}($

$< d_{Ffrom}: \mathbf{H}|0 \leq d_{Ffrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Fto}: \mathbf{H}|0 \leq d_{Fto}\mathbf{H} \leq FF\mathbf{H} >$

$< MatrixP_{Detect}: \mathbf{H}\big|\big[d_{Ffrom}\big][d_{Fto}] >\big) >)$ (Eq-4.23f)

The Reachability tool informs about the reachability of certain modes of the system, i.e., $[A - \lambda I \quad B]$ has full row rank or the entire system is reachable [4.2] provided that $C = [B \quad AB \quad A^2B\, A^3B \quad A^4B \quad A^5B \ldots . A^{n-1}B]$ has full rank and the corresponding architectural infrastructure includes:

$ToolReachability\mathbf{ST} = \quad ($

$< InputInterface\mathbf{ST}($

$< d_{Afrom}: \mathbf{H}|0 \leq d_{Afrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Ato}: \mathbf{H}|0 \leq d_{Ato}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Bfrom}: \mathbf{H}|0 \leq d_{Bfrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Bto}: \mathbf{H}|0 \leq d_{Bto}\mathbf{H} \leq FF\mathbf{H} >$

$< InPointerMatrixA: \mathbf{H}|[d_{Afrom}][d_{Ato}] >$

$< InPointerMatrixC: \mathbf{H}|[d_{Bfrom}][d_{Bto}] >$

$< OutputInterface\mathbf{ST}($

$< OutRankMatrixP_{Contr}: \mathbf{H}|0 \leq OutRankMatrixP_{Contr} \leq FF\mathbf{H} >$

$< ProcessingSpace\mathbf{ST}($

$< d_{Cfrom}: \mathbf{H}|0 \leq d_{Cfrom}\mathbf{H} \leq FF\mathbf{H} >$

$< d_{Cto}: \mathbf{H}|0 \leq d_{Cto}\mathbf{H} \leq FF\mathbf{H} >$

$$< MatrixP_{Reachab}\colon \mathbf{H}\left|\left[d_{Cfrom}\right]\left[d_{Cto}\right] >\right) >) \qquad \text{(Eq-4.23g)}$$

Thus, the availability of the above tools provide the capability to test the system's status.

4.2.2 Applied Control

The model's self-controlling component forms a loop around the layered component while the constraints component is cascaded within this loop. The described infrastructure presents an intrinsic adaptive control to the sensed parameters which is supported by a knowledge base. In Figure-4.9 it is depicted the formed first-order control loop which is described by the relation of (Eq-8.2) which can be re-written as

$$\begin{aligned}
&HomeUbiHealth\S.Architecture\mathbf{ST} \triangleq \{\\
&\quad < Process\mathbf{ST}(\\
&\qquad < (< {}^{TopLevel}_{i\mathbf{N}=\mathbf{0}}R(Level[i\mathbf{N}]\mathbf{ST} >\\
&\qquad\quad \| < (< MedicalConstraints\mathbf{ST} >\\
&\qquad\quad \| < HealthStatus\mathbf{ST} >\\
&\qquad\quad \| < PersonalPreferences\mathbf{ST} >) >\\
&\quad < RuleBased\mathbf{ST} >\\
&\qquad \| < {}^{Taxonomies}_{k\mathbf{N}=\mathbf{0}}RKnowledge(k\mathbf{N})\mathbf{ST} >) >\\
&\quad < EventDriver\mathbf{ST} >) >\\
&\quad \| < {}^{Instrument}_{j\mathbf{N}=\mathbf{0}}R(Tool(j\mathbf{N})\mathbf{ST}) > \} \qquad \text{(Eq-4.24)}
\end{aligned}$$

The corresponding block diagram of the architectural infrastructure is presented in Figure-4.9.

The self-controlling component is acting as an adaptive controller, see Figure-4.2, aiming at providing guidance-instructions to the physical devices such that the model satisfies the stability criteria set by the medical professionals. According to Control Systems theory, a system is stable if the output is bounded for all bounded inputs. In our case, the inputs to the model are constrained by the $Constraints\mathbf{ST}$ component applying medical constraints, health status, and personal preferences. Taking advantage of the Routh Array criterion, an effective test can be performed

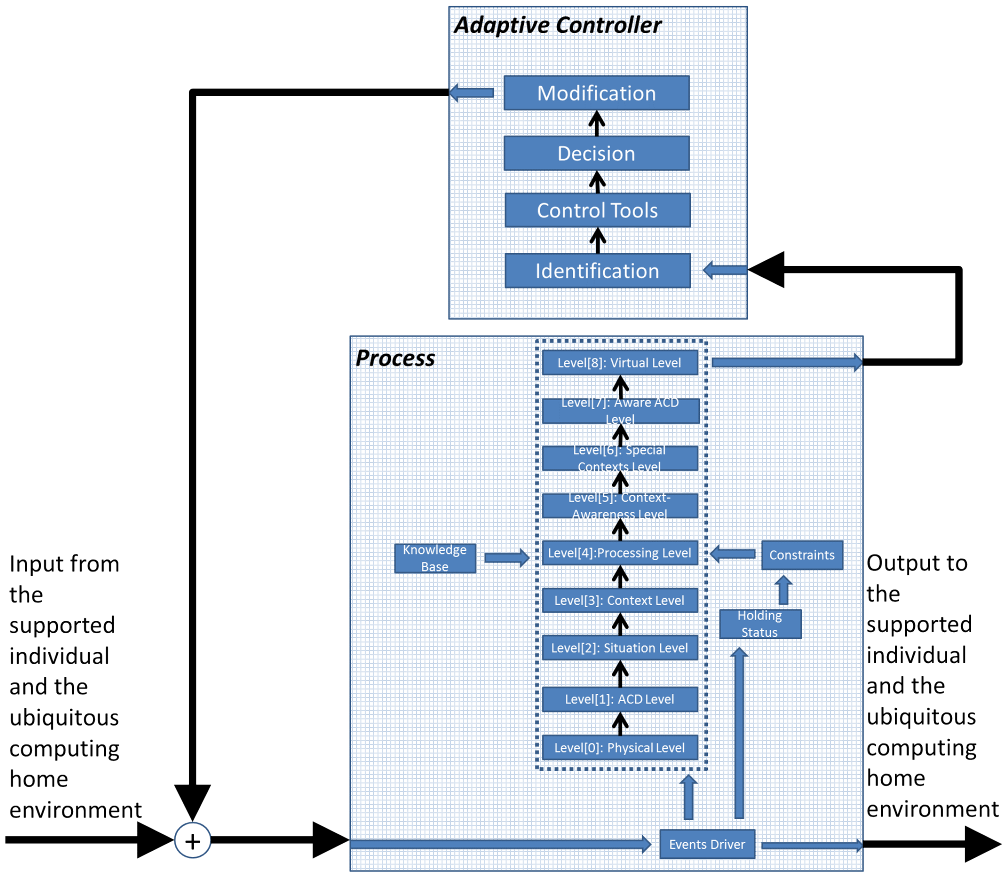

Adaptive Controller
Modification
Decision
Control Tools
Identification
Process
Level[8]: Virtual Level
Level[7]: Aware ACD Level
Level[6]: Special Contexts Level
Level[5]: Context-Awareness Level
Knowledge Base
Level[4]:Processing Level
Constraints
Level[3]: Context Level
Holding Status
Level[2]: Situation Level
Level[1]: ACD Level
Level[0]: Physical Level
Events Driver
Input from the supported individual and the ubiquitous computing home environment
+
Output to the supported individual and the ubiquitous computing home environment

Figure-4.9 The Home UbiHealth model as a first-order loop control system.

requiring the coefficients of the first column of the controller to be all positive. Other stability tests exist such as the Bode and the Nyquist stability criterion.

4.3 Tools for self-control

The behavior of the infrastructure depends on the self-control component which applies control on three directions:

1) Control on Concepts: the involved concepts at the virtual layer of the infrastructure form a system of concepts which is bounded by the provided values and limits from the constraining component of the infrastructure. The designed architecture forms a first order control loop with the input provided from the physical layer and the dispersed sensing devices, the input is bounded by the constrains components while the output is also bounded by the constrains component in order to obtain a stable system. The controlled system is consisted of the concepts involved in the virtual layer.

2) Action Control: the self-control component provides the output to the actuating devices at the physical layer through the Events Driver component. The output of the self-control component is directed to the events driver component which, in turn, advances the controlled values at the specified actuating devices.
3) Applications Migration: the self-control component provides the required information for the synchronization of the migration of applications from a device to another at the physical layer. The applications migration is synchronized in order to support the individual's mobility locating and identifying the departing and the destination devices for the application's transfer.

The internal structure of the self-control component must be capable of accommodating the infrastructure for a complete fist-order loop control system, the buffers for the provided output from the control system along with synchronizing information for applications' migration to support the individual's mobility.

4.3.1 Control on Concepts

For the development of a system it is required:

1. Input: it is provided by the sensing devices at the physical layer enriched by the constraints provided by the medical professionals, the personal preferences of the individual, and the individual's health status, as well as, the additional attributes and properties associated with each device or the relations among the devices by the supporting taxonomies.
2. Output: directed to the Events Driver to supply the actuating devices at the physical layer. The output of the system consists of values to be fed at the physical devices and values to feed artificial constructs assisting in the development of the virtual layer.
3. States variables: the set of variables that describe the discrete states of the system allowing the estimation of the next state given the current values of the state variables. In other words, the next state of the system depends on the interactions with current values while the set of state variables characterizes the system.

In Figure-4.10 it is presented the block diagram of a system showing the input, the output, the initial values and the state of the system. The state of the system is

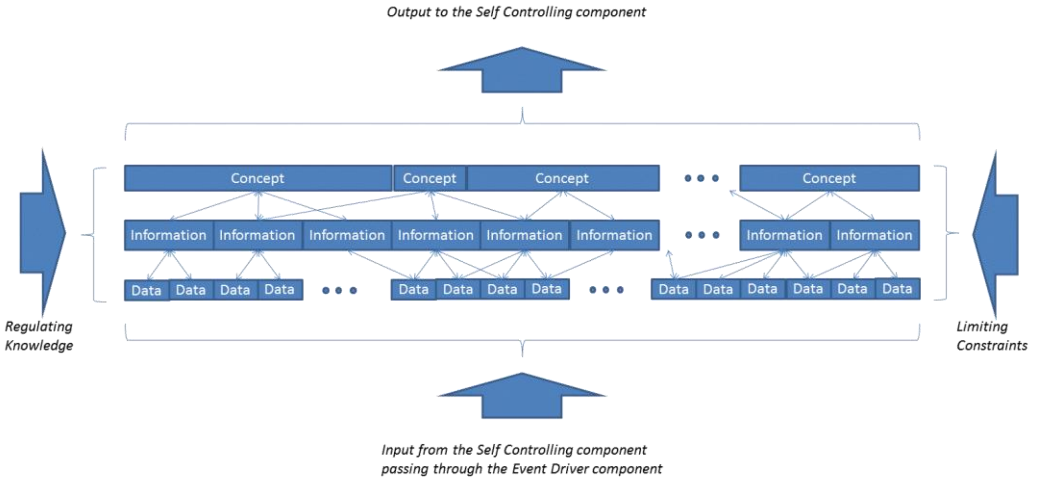


Figure-4.10 The controlled system's elements and the regulating constraints.

represented by a set of variables $X = \{x_1, x_2, x_3, \ldots\ldots, x_n\}$ which are interacting and provide a square matrix $\bar{A} = X\mathbf{x}X$ representing the state matrix, a n x n matrix. The set of inputs $U = \{u_1, u_2, u_3, \ldots .\, u_m\}$ is interacting with the state variables forming a matrix $\bar{B} = X$ x U which is a nxm matrix. Hence, the differential equation $\dot{x} = \bar{A}x + \bar{B}u$ describes the change of the system's state. The set of outputs from the system $Y = \{y_1, y_2, y_3, \ldots .\, y_k\}$ are related with state variables $X = \{x_1, x_2, x_3, \ldots\ldots, x_n\}$ forming the output matrix $\bar{C} = X\mathbf{x}Y$ which is a n x k matrix and the output related with the input form the $\bar{D} = Y$ **x** U which is a k x m matrix. Hence, the linear equation $y = \bar{C}x + \bar{D}u$ describes the output of the system as a function of the provided input and the state variables. The system is described by the following set of equations:

$$\dot{x} = \bar{A}x + \bar{B}u$$

$$y = \bar{C}x + \bar{D}u \qquad \text{(Eq-4.26)}$$

The design requires three phases:

- Phase-1: consider the entire set of state variables to be fed back into the virtual layer, each state is related directly with the corresponding concept of the virtual layer.
- Phase-2: construct an "observer" for those states that are not directly sensible and not directly present in the output.
- Phase-3: connect the "observer" to the feedback.

The examination of the behavior of the system is performed with the tools of

- Controllability: calculating the rank of the square matrix $P_c = [B \quad AB \quad A^2B \dots . A^{n-1}B]$
- Observability: calculating the determinant of the matrix $P_o = \begin{bmatrix} C \\ CA \\ CA^{-1} \\ \vdots \\ CA^{n-2} \\ CA^{n-1} \end{bmatrix} \neq 0$

With the above tools the state of the system can be controlled which implies that the system of concepts can be controlled and hence, the information and the related data can be controlled. In other words, controlling the interacting concepts, the associated information and then, the constituting data can be controlled too. The applying prerequisite refers to the inclusion of concepts of the same ontological abstraction or level. When there is a conflict or an ambiguity arises in the control of the system, then concepts from a lower ontological level must be used to continue with conceptual consistency. Likewise, when the system's constituting concepts have been defined, the availability of concepts at a higher ontological level must be examined to reduce the number of system's variables. The relation (Eq-4.19) contains the component $< Controller\mathbf{ST} >$ which includes the structural elements of

$Controller\mathbf{ST} = ($

$< MatrixA\mathbf{ST} >$

$< MatrixB\mathbf{ST} >$

$< MatrixC\mathbf{ST} >$

$< MatrixD\mathbf{ST} >$

$< InputUvector\mathbf{ST} >$

$< OutputYvector\mathbf{ST} >$

$< StateXvector\mathbf{ST} > \quad)$ (Eq-4.27)

The feedback loop is developed through the physical layers of the infrastructure in order to reconsider the relations among the involved state variables. Also, the feedback is passing through the layered infrastructure in order to reform the mixture and the values of the involved concepts. The change in the relations and the corresponding values of the relations cause the corresponding changes to the values of the actuating devices related to the changing values of the involved concepts. Figure-4.10 depicts the system consisted of interrelated data, information, and concepts along with the limiting constraints and the regulating knowledge. Hence, a change in the state of a concept requires the change in the state values of the constituting information and the supporting data causing a restructuring of values and relations.

4.3.2 Action Control

The Self-Control component outputs the values intended for the actuating devices. The control is applied on the virtual corresponding layer of the model's infrastructure that contains the physical, the virtual components, and their identified relationships. The control is performed by the self-controlling component that must provide the structural facility to accommodate the identification for each physical actuating device, the related decision made by the self-controlling component considering the entire situation of the model, and the action to be taken for the specific actuating device. From the relation (Eq-4.19) results the term $< OptimalDecision\mathbf{ST} >$ in more details

$$\begin{aligned} OptimalDecision\mathbf{ST} = (\ & < Identification\mathbf{ST} > \\ & < Decision\mathbf{ST} > \\ & < Adaptation\mathbf{ST} >) \end{aligned} \qquad \text{(Eq-4.28)}$$

The relation (Eq-4.28) provides the means to obtain the closed control loop of the model. Figure-4.11 illustrates the formed loop with a self-controlling component accommodating the identity, the taken decision, and the adaptation values.

The sensed raw data is passed through the Event Driver, it is processed by the layered infrastructure and then it is represented in the formed virtual layer where there is a complete representation of the holding situation at the physical environment. Then, the self-controlling component identifies the involved device(s), it decides upon evidences, and adapts the value that must be provided to the actuating device.

4.3.3 Applications Migration

The developing situations in the ubiquitous computing environment require forcing the abrupt interruption of the software applications operation. There are many various reasons for sudden discontinuities in the software applications operation such as power failures, introduction of electromagnetic interference, white noise, or wireless

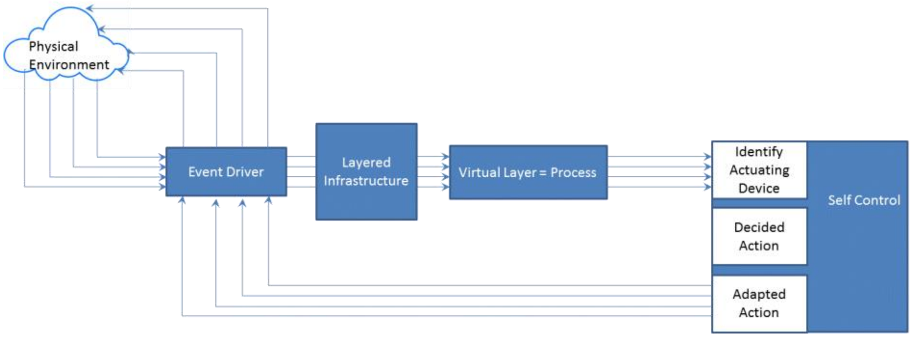


Figure-4.11 The first-order control loop of the controlled Process.

connectivity limitations. The model must be capable or realizing, facing, and responding to interrupts finding alternatives in order to keep supporting transparently the uninterruptable operation of the system. The model autonomously, without any external intervention, adapts its operation in order to achieve the set goals and aims. The self-controlling component must provide the proper facilities to accommodate the necessary information to coordinate and synchronize the operations of migration, assisting the transfer of a software application that satisfies the operational conditions.

The examination of the relation (Eq-4.28) reveals that the component must involve the capability to contain the required information for migration. Each of the three stages of (Eq-4.28) must be as it follows:

$OptimalDecision\mathbf{ST} = ($

$< Identification\mathbf{ST}($

$< Modification\mathbf{ST}($

$< Modify{:}\,\mathbf{BL}|Modify\mathbf{BL} = \{(T,on),(F,off)\} >$

$\| < ModifyDevice\mathbf{ST} >) >$

$< Migration\mathbf{ST}($

$< Migrate{:}\,\mathbf{BL}|Migrate\mathbf{BL} = \{(T,on),(F,off)\} >$

$\| < MigratingDevices\mathbf{ST}($

$< MigrateFromDevice\mathbf{ST} >$

$< MigrateToDevice\mathbf{ST} >) >) >) >$

$< Decision\mathbf{ST}($

$< DecidedActions\mathbf{ST} >$

$< DecidedDevices\mathbf{ST} >) >$

$< Adaptation\mathbf{ST}($

$$< AdaptedPlan\mathbf{ST} >$$
$$< AdaptedDevices\mathbf{ST} >$$
$$< AdaptedActions\mathbf{ST} >) > \ ) \qquad \text{(Eq-4.29)}$$

For the case of migration, the $Migrate\mathbf{BL}$ component must be set to $True$ and then the involved devices must be identified with the components of $MigrateFromDevice\mathbf{ST}$ and the $MigrateToDevice\mathbf{ST}$. Then, the decided action plan must be accommodated in the data structures of the $Decision\mathbf{ST}$ component and the information related to the involved activities must be placed in the $Adaptation\mathbf{ST}$ component.

4.4 Static Behavior: Transform Data to Concepts

The static behavior refers to those components of the design that are known at compilation time. In other words, the static behavior section contains the descriptions of the functional components of the designed model. The architectural components have functional characteristics with which is accomplished the flow of data within the described architecture. The interaction of those functional characteristics causes the propagation of data within the architectural infrastructure. Hence, the static behavior is determined from the characteristic capabilities presented by the architectural elements that achieve the transfer and transformation of the administered data.

The interactions among the architectural elements of the infrastructure represent the static specifications of the model and they are known at design and compilation time. The functionality of the architectural elements must present an input and an output handle to achieve the interactions with other coexisting architectural elements along with a processing component. The architectural elements characteristics are defined through the employment of Unified Data Models (UDM) and Unified Process Models (UPM) [4.3]. The UDM structure is used to describe the architectural elements while the UPM structure is used to describe the static behavior of the architectural elements. The general form of UDM as it is described in [4.3] is given by $\tau =< ElementId{:}\, type \mid (ApplyingConstraints) >$ with a formal definition from [4.3] as

$$UDM \triangleq \tau =< {}_{i\mathbf{N}=1}^{n\mathbf{N}}R < S_i | \forall e_i \in S_i, p_i(e_i) > \qquad \text{(Eq-4.30)}$$

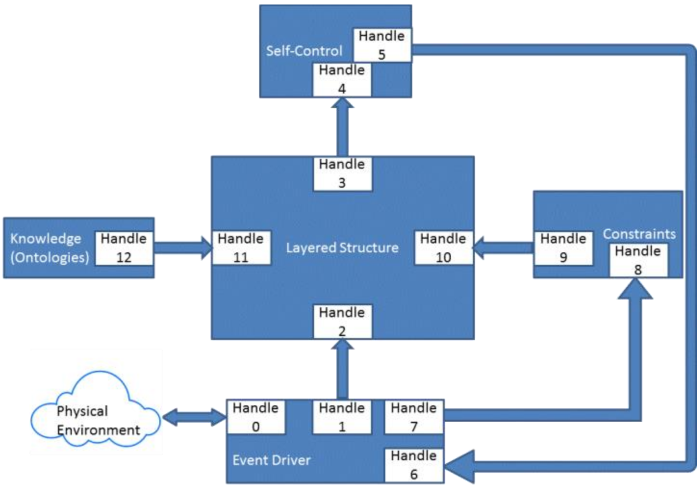


Figure-4.12 Block diagram of the Home UbiHealth model's internal data handles.

where, p_i denotes some arbitrary shared property among the elements of the S_i set. The UPM construct is used to describe the behavior of a software program $\wp$ governed by logical operations ruled by interrupting events occurring in the dimension of time which is formally described in [4.3] as a dispatching procedure:

$$\wp = {}_{k=1}^{m}R(@e_k \Lsh P_k) = {}_{k=1}^{m}R\left[@e_k \Lsh {}_{i=1}^{n-1}R\left(s_i(k)r_{ij}(\boldsymbol{k})s_j(k)\right)\right] \qquad \text{(Eq-4.31)}$$

$with\, j = i + 1$ and the term ${}_{i=1}^{n-1}R\left(s_i(k)r_{ij}(\boldsymbol{k})s_j(k)\right)$ denoting the involvement of process P_k for the flow of program's operations. Considering the availability of UDM and UPM constructs, then, the static behavior of the model's architecture can be defined using the defined framework of Real-Time Process Algebra (RTPA) defined in [4.4]. The RTPA framework can be used to describe models of conceptual systems which are in between their logical and physical counterparts. The RTPA framework is defined by the triplet of $RTPA \triangleq (\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{P}}, \boldsymbol{\mathcal{N}})$ where $\boldsymbol{\mathcal{T}}$ representing 17 primitive types to model systems, $\boldsymbol{\mathcal{P}}$ representing 17 primitive meta-processes to model the behavior of systems, and $\boldsymbol{\mathcal{N}}$ representing 17 primitive relations among processes to model complex systems' behaviors.

In the following paragraphs it is given the transformation process of raw data to the development of the virtual environment based on the constraints and the available medical knowledge level. Among the set constraints are the personal preferences of the individual whose evolving status of health is controlled by the described Home UbiHealth model. The model of the formed system includes the supported individual whose physiological and psychological aspects of the health status are considered and examined by the existing knowledge level at the virtual layer of the described model.

The Home UbiHealth model involves a layered structure with which it processes the acquired raw data, at the lowest level, in order to produce, at the highest level, the virtual corresponding model of the physical environment. Recalling the relations (Eq-4.21a) through (Eq-4.21h) it is obtained the gradual transformation of raw data. Figure-4.12 depicts the building blocks of the data transformation component along with the required handles with the supporting components of Event Driver, Knowledge, Constraints, and Self-Control.

The static behavior of the data flows in the Home UbiHealth model is described by the following relation (Eq-4.32) emphasizing on the use of the involved data handles that ensures the functionality among the involved main system's components,

$< FlowDataOperation\mathbf{PC} = ($

$\quad EventDriverFlowData($

$\qquad < \mathbf{I} :: (InRawData\mathbf{TYPE}, OutControlData\mathbf{TYPE}) >$

$\qquad < \mathbf{O} :: \left(\begin{matrix} OutRawData\mathbf{TYPE}, OutProcessedData\mathbf{TYPE}, Medical\mathbf{TYPE}, \\ Preferences\mathbf{TYPE},\ HealthStatus\mathbf{TYPE} \end{matrix}\right) >$

$\qquad < \mathbf{UDM} ::<$

$\qquad EventDriver\mathbf{ST}, Handle0\mathbf{ST}, Handle1\mathbf{ST}, Handle6\mathbf{ST}, Handle7\mathbf{ST} >)\mathbf{PC}$

$\quad LayersFlowData($

$\qquad < \mathbf{I} :: (OutRawData\mathbf{TYPE},\ Constraint\mathbf{TYPE},\ Knowledge\mathbf{TYPE}) >$

$\qquad < \mathbf{O} :: (VirtualData\mathbf{TYPE}) >$

$\qquad < \mathbf{UDM} ::< Layer1\mathbf{ST}, Layer2\mathbf{ST}, Layer3\mathbf{ST}, Layer4\mathbf{ST}, Layer5\mathbf{ST},$

$\qquad Layer6\mathbf{ST}, Layer7\mathbf{ST}, Layer8\mathbf{ST}, Layer9\mathbf{ST}, Handle2\mathbf{ST}, Handle3\mathbf{ST},$

$\qquad Handle10\mathbf{ST}, Handle11\mathbf{ST} >)\mathbf{PC}$

$\quad ConstraintsFlowData($

$\qquad < \mathbf{I} :: (Medical\mathbf{TYPE},\ Preferences\mathbf{TYPE},\ HealthStatus\mathbf{TYPE}) >$

$\qquad < \mathbf{O} :: (Constriant\mathbf{TYPE}) >$

$\qquad < \mathbf{UDM} ::< RuleBasedSystem\mathbf{ST}, Handle8\mathbf{ST}, Handle9\mathbf{ST} >)\mathbf{PC}$

$\quad KnowledgeFlowData($

$\qquad < \mathbf{I} :: (Query\mathbf{TYPE}) >$

$\qquad < \mathbf{O} :: (Knowledge\mathbf{TYPE}) >$

$< \mathbf{UDM} ::< \mathit{RelationResolution}\mathbf{ST}, \mathit{Handle}12\mathbf{ST} >)\mathbf{PC}$

$\mathit{SelfControlFlowData}($

$< \mathbf{I} :: (\mathit{VirtualData}\mathbf{TYPE}) >$

$< \mathbf{O} :: (\mathit{OutControlData}\mathbf{TYPE}) >$

$< \mathbf{UDM} ::< \mathit{ControlInfrastructure}\mathbf{ST}, \mathit{Handle}5\mathbf{ST}, \mathit{Handle}6\mathbf{ST} >)\mathbf{PC}$

$) >$ (Eq-4.32)

The distinguished identification of the handles provides the capability to perform matching of the data exchanged among the major components of the model. The output of any two cooperating handles is going to be input to the other and hence, pairs of handles must present matching input-output data structures and the corresponding data types. The output of each of the used handles is either input to its cooperating corresponding handle or output to the contained model's component.

4.4.1 Supporting Handles

The assigned handles provide the architectural infrastructure for the operation of the designed model. The description of each of the 13 handles is provided with the following relations:

$< \mathit{Handle}0\mathbf{ST}($

$< \mathit{Input}0{:}\, \mathbf{TYPE0} | (\mathbf{TYPE0} \sqsubseteq \mathbf{TYPE}) \wedge (\mathit{Input}0 \in \mathbf{TYPE0}) >$

$< \mathit{Output}0{:}\, \mathbf{TYPE0} | (\mathbf{TYPE0} \sqsubseteq \mathbf{TYPE}) \wedge (\mathit{Input}0 \in \mathbf{TYPE0}) >) >$

$< \mathit{Handle}1\mathbf{ST}($

$< \mathit{Output}0{:}\, \mathbf{TYPE0} | (\mathbf{TYPE0} \sqsubseteq \mathbf{TYPE}) \wedge (\mathit{Output}0 \in \mathbf{TYPE0}) >$

$< \mathit{Output}1{:}\, \mathbf{TYPE1} | (\mathbf{TYPE1} \sqsubseteq \mathbf{TYPE}) \wedge (\mathit{Output}1 \in \mathbf{TYPE1}) >) >$

$< \mathit{Handle}2\mathbf{ST}($

$< \mathit{Output}1{:}\, \mathbf{TYPE1} | (\mathbf{TYPE1} \sqsubseteq \mathbf{TYPE}) \wedge (\mathit{Output}1 \in \mathbf{TYPE1}) >$

$< \mathit{Output}2{:}\, \mathbf{TYPE2} | (\mathbf{TYPE2} \sqsubseteq \mathbf{TYPE}) \wedge (\mathit{Output}2 \in \mathbf{TYPE2}) >) >$

$< \mathit{Handle}3\mathbf{ST}($

$< \mathit{Input}3{:}\, \mathbf{TYPE3} | (\mathbf{TYPE3} \sqsubseteq \mathbf{TYPE}) \wedge (\mathit{Input}3 \in \mathbf{TYPE3}) >$

$< \mathit{Output}3{:}\, \mathbf{TYPE3} | (\mathbf{TYPE3} \sqsubseteq \mathbf{TYPE}) \wedge (\mathit{Output}3 \in \mathbf{TYPE3}) >) >$

$< \mathit{Handle}4\mathbf{ST}($

$< \mathit{Output}3{:}\, \mathbf{TYPE3} | (\mathbf{TYPE3} \sqsubseteq \mathbf{TYPE}) \wedge (\mathit{Output}3 \in \mathbf{TYPE3}) >$

$< \mathit{Output}4{:}\, \mathbf{TYPE4} | (\mathbf{TYPE4} \sqsubseteq \mathbf{TYPE}) \wedge (\mathit{Output}4 \in \mathbf{TYPE4}) >) >$

$$
\begin{array}{l}
< Handle5\mathbf{ST}(\\
\qquad < Input5\colon \mathbf{TYPE5}|(\mathbf{TYPE5} \sqsubseteq \mathbf{TYPE})\wedge(Input5 \in \mathbf{TYPE5}) > \\
\qquad < Output5\colon \mathbf{TYPE5}|(\mathbf{TYPE5} \sqsubseteq \mathbf{TYPE})\wedge(Output5 \in \mathbf{TYPE5}) >) > \\
< Handle6\mathbf{ST}(\\
\qquad < Output5\colon \mathbf{TYPE5}|(\mathbf{TYPE5} \sqsubseteq \mathbf{TYPE})\wedge(Output5 \in \mathbf{TYPE5}) > \\
\qquad < Output6\colon \mathbf{TYPE6}|(\mathbf{TYPE6} \sqsubseteq \mathbf{TYPE})\wedge(Output6 \in \mathbf{TYPE6}) >) > \\
< Handle7\mathbf{ST}(\\
\qquad < Input7\colon \mathbf{TYPE7}|(\mathbf{TYPE7} \sqsubseteq \mathbf{TYPE})\wedge(Input7 \in \mathbf{TYPE7}) > \\
\qquad < Output7\colon \mathbf{TYPE7}|(\mathbf{TYPE7} \sqsubseteq \mathbf{TYPE})\wedge(Input7 \in \mathbf{TYPE7}) >) > \\
< Handle8\mathbf{ST}(\\
\qquad < Output7\colon \mathbf{TYPE7}|(\mathbf{TYPE7} \sqsubseteq \mathbf{TYPE})\wedge(Output7 \in \mathbf{TYPE7}) > \\
\qquad < Output8\colon \mathbf{TYPE8}(\mathbf{TYPE8} \sqsubseteq \mathbf{TYPE})\wedge(Output8 \in \mathbf{TYPE8}) >) > \\
< Handle9\mathbf{ST}(\\
\qquad < Input9\colon \mathbf{TYPE9}|(\mathbf{TYPE9} \sqsubseteq \mathbf{TYPE})\wedge(Input9 \in \mathbf{TYPE9}) > \\
\qquad < Output9\colon \mathbf{TYPE9}|(\mathbf{TYPE9} \sqsubseteq \mathbf{TYPE})\wedge(Output9 \in \mathbf{TYPE9}) >) > \\
< Handle10\mathbf{ST}(\\
\qquad < Output9\colon \mathbf{TYPE9}|(\mathbf{TYPE9} \sqsubseteq \mathbf{TYPE})\wedge(Output9 \in \mathbf{TYPE9}) > \\
\qquad < Output10\colon \mathbf{TYPE10}|(\mathbf{TYPE10} \sqsubseteq \mathbf{TYPE})\wedge(Output10 \in \mathbf{TYPE10}) >) > \\
< Handle11\mathbf{ST}(\\
\qquad < Input11\colon \mathbf{TYPE11}|(\mathbf{TYPE11} \sqsubseteq \mathbf{TYPE})\wedge(Input11 \in \mathbf{TYPE11}) > \\
\qquad < Output11\colon \mathbf{TYPE11}|(\mathbf{TYPE11} \sqsubseteq \mathbf{TYPE})\wedge(Output11 \in \mathbf{TYPE11}) >) > \\
< Handle12\mathbf{ST}(\\
\qquad < Output11\colon \mathbf{TYPE11}|(\mathbf{TYPE11} \sqsubseteq \mathbf{TYPE})\wedge(Output11 \in \mathbf{TYPE11}) > \\
\qquad\; < Output12\colon \mathbf{TYPE12}|(\mathbf{TYPE12} \sqsubseteq \mathbf{TYPE})\wedge(Output12 \in \mathbf{TYPE12}) >) \\
>
\end{array}
\tag{Eq-4.33}
$$

At each construct, the first field of each pair refers to input while the second field refers to the corresponding output. Also, any two cooperating handles present the same input-output pairs keeping the same data type. For each data transfer, a dedicated data type is used that corresponds to the internal structure of the transferring data.

4.4.2 Layered Structure and Data Traceability

The core internal structure of the model is consisted of a set of layers. Each layer receives a set of values and after proper processing produces a transformed set of data. The transforming functionality is analogous to the reference brain model referred to [4.5] performing both subconscious and conscious transformation on the received inputs. Analogously to the brain reference model, the structure starts with the sensation of the external causes, the storage to memory of the sensed signal, the perception based on knowledge and past experiences, the initiation of activities responding subconsciously to the external signals, the performance of meta-cognitive functioning which means the interrelation of the received signals with other experiences stored in memory, the inference from the meta-cognitive functioning forming meta-inferences which means inferences on higher level conceptual constructs, and finally, the performance of higher cognitive functioning.

In the case of the Home UbiHealth model an analogous layered processing is employed based on the given architectural infrastructure. Each layer contains a memory structure to accommodate the corresponding data set applying the proper each time transforming processes. The typical structure of a layer consists of the input, the processing, the output, and the processing memory segments of the applying static operation of a software code that performs the desired transformation, i.e.,

$$
\begin{array}{ll}
< LayerName\mathbf{ST}(& \\
\quad < \mathbf{I} :: (InputToLayer\mathbf{TYPE}\boldsymbol{\alpha}|(\text{Applying constraints})) > & \\
\quad < \mathbf{O} :: (OutputFromLayer\mathbf{TYPE\beta}|(Applying\ constraints)) > & \\
\quad < \mathbf{UDM} ::< (TransformingProgramCode\mathbf{PC} > \ ) & \text{(Eq-4.34)}
\end{array}
$$

At each layer, the element $InputToLayer\mathbf{TYPE}\boldsymbol{\alpha}$ refers to a data structure of type $\mathbf{TYPE}\alpha$ which must be equivalent to the type and structure of the output element of the previous layer at a lower level. Similarly, the element's $OutputFromLayer\mathbf{TYPE\beta}$ type and structure must be equivalent to the input element of the next, higher level, layer. The element $TransformingProgramCode\mathbf{PC}$ performs the actual transformation processing the receive data in order to obtain the desired structures of the output that is required to match with the input of the higher level layer. Analyzing the element $LayersFlowData$ of (Eq-4.32) contains:

$$\mathit{LayeredStructure}\mathbf{PC} =$$

$$($$

$$\mathit{Layer}1\mathit{PhysicalLevel}($$

$$< \mathbf{I} :: (\mathit{OutputOfEventsDriver}\mathbf{TYPE}\boldsymbol{\alpha}) >< \mathbf{O} ::$$

$$(\mathit{ACDStructure}\mathbf{TYPE}\boldsymbol{\beta}) >< \mathbf{UDM} ::< \mathit{Handle}2\mathbf{ST}, \mathit{Layer}1\mathbf{ST} >)\mathit{PC}$$

$$\mathit{Layer}2\mathit{ACDLevel}($$

$$< \mathbf{I} :: (\mathit{ACDStructure}\mathbf{TYPE}\boldsymbol{\beta}) >< \mathbf{O} :: (\mathit{Situation}\mathbf{TYPE}\boldsymbol{\gamma}) >< \mathbf{UDM} ::<$$

$$\mathit{Layer}2\mathbf{ST} >)\mathit{PC}$$

$$\mathit{Layer}3\mathit{SituationLevel}($$

$$< \mathbf{I} :: (\mathit{Situation}\mathbf{TYPE}\boldsymbol{\gamma}) >< \mathbf{O} :: (\mathit{Context}\mathbf{TYPE}\boldsymbol{\delta}) >< \mathbf{UDM} ::<$$

$$\mathit{Layer}3\mathbf{ST} >)\mathit{PC}$$

$$\mathit{Layer}4\mathit{ContextLevel}($$

$$< \mathbf{I} :: (\mathit{Context}\mathbf{TYPE}\boldsymbol{\delta}) >< \mathbf{O} :: (\mathit{Processing}\mathbf{TYPE}\boldsymbol{\varepsilon}) >< \mathbf{UDM} ::<$$

$$\mathit{Layer}4\mathbf{ST} >)\mathit{PC}$$

$$\mathit{Layer}5\mathit{ProcessingLevel}($$

$$< \mathbf{I} :: (\mathit{Processing}\mathbf{TYPE}\boldsymbol{\varepsilon}) >< \mathbf{O} :: (\mathit{ContextAware}\mathbf{TYPE}\boldsymbol{\zeta}) >< \mathbf{UDM} ::$$

$$< \mathit{Handle}10\mathbf{ST}, \mathit{Handle}11\mathbf{ST}, \mathit{Layer}5\mathbf{ST}, \mathit{Constraint}\mathbf{ST}, \mathit{Knowledge}\mathbf{ST} >$$

$$)\mathit{PC}$$

$$\mathit{Layer}6\mathit{ContextAwareLevel}($$

$$< \mathbf{I} :: (\mathit{Processing}\mathbf{TYPE}\boldsymbol{\zeta}) >< \mathbf{O} :: (\mathit{ContextAware}\mathbf{TYPE}\boldsymbol{\eta}) >< \mathbf{UDM} ::$$

$$< \mathit{Handle}11\mathbf{ST}, \mathit{Layer}6\mathbf{ST}, \mathit{Knowledge}\mathbf{ST} >)\mathit{PC}$$

$$\mathit{Layer}7\mathit{SpecialContextLevel}($$

$$< \mathbf{I} :: (\mathit{ConstextAware}\mathbf{TYPE}\boldsymbol{\eta}) >< \mathbf{O} :: (\mathit{SpecialContext}\mathbf{TYPE}\boldsymbol{\theta}) ><$$

$$\mathbf{UDM} ::< \mathit{Handle}11\mathbf{ST}, \mathit{Layer}7\mathbf{ST}, \mathit{Knowledge}\mathbf{ST} >)\mathit{PC}$$

$$\mathit{Layer}8\mathit{AwareACDLevel}($$

$$< \mathbf{I} :: (\mathit{SpeciaContex}\mathbf{TYPE}\boldsymbol{\theta}) >< \mathbf{O} :: (\mathit{AwareACD}\mathbf{TYPE}\boldsymbol{\iota}) >< \mathbf{UDM} ::<$$

$$\mathit{Handle}11\mathbf{ST}, \mathit{Layer}8\mathbf{ST}, \mathit{Knowledge}\mathbf{ST} >)\mathit{PC}$$

$$\mathit{Layer}9\mathit{VirtualLevel}($$

$$(< \mathbf{I} :: (\mathit{AwareACD}\mathbf{TYPE}\boldsymbol{\iota}) >< \mathbf{O} :: (\mathit{Virtual}\mathbf{TYPE}\boldsymbol{\kappa}) >< \mathbf{UDM} ::<$$

$$\mathit{Handle}3\mathbf{ST}, \mathit{Handle}11\mathbf{ST}, \mathit{Layer}8\mathbf{ST}, \mathit{Knowledge}\mathbf{ST} >)\mathbf{PC}$$

$$)\mathbf{PC} \qquad \text{(Eq-4.35)}$$

The software application that implements the operation of the layered structure requires the inclusion of the data structures and the software program codes described in (Eq-4.35) which describes the static behavior of the required software application. At the fifth layer are introduced the constraints and the required ontologies, and from the fifth layer and on, the ontologies support the performed processing by augmenting the available relations among data structures.

The acquired data from the physical layer enter the model and participate in processing which can alter, reform, or create additional data to represent information or conceptual structures. The initially employed data follow a processing path that leads to the development of meta-cognitive concepts at the virtual layer which corresponds to the augmented reality of the physical layer. The processes of validation and verification require the capability to represent the initially acquired data in the resulting decisions. In order to achieve tracking the data of the layered structure either from the entrance or the exit, the developed types of the developed and involved data structures must satisfy the following pre-condition:

$$\mathbf{TYPE}\boldsymbol{\alpha} \sqsubseteq \mathbf{TYPE}\boldsymbol{\beta} \sqsubseteq \mathbf{TYPE}\boldsymbol{\gamma} \sqsubseteq \mathbf{TYPE}\boldsymbol{\delta} \sqsubseteq \mathbf{TYPE}\boldsymbol{\varepsilon} \sqsubseteq \mathbf{TYPE}\boldsymbol{\zeta} \sqsubseteq \mathbf{TYPE}\boldsymbol{\eta} \sqsubseteq \mathbf{TYPE}\boldsymbol{\theta} \sqsubseteq \mathbf{TYPE}\boldsymbol{\iota} \sqsubseteq \mathbf{TYPE}\boldsymbol{\kappa} \sqsubseteq \mathbf{TYPE} \qquad \text{(Eq-4.36)}$$

The relation of the involved types in (Eq-4.36a) reveals that the contribution of a single variable can be traced in the development of its virtual counterpart by replacing $Handle3$**ST** with $A1$ and *Layer9***ST**, *Layer8***ST**, *Layer7***ST**, *Layer6***ST**, *Layer5***ST**, *Layer4***ST**, *Layer3***ST** as well as $Layer2$**ST**, *and* $Layer1$**ST** with $A2,$ substituting $Hand2$**ST** with $A3,$ replacing the component *Handle1***ST**.*EventDriver***ST**.*handle0***ST** with $A4,$ then, the contribution at the obtained meta-cognitive level can be determined,

$$A1.A2.A3.A4.(variable\mathbf{TYPEa}) = value \qquad \text{(Eq-4.36a)}$$

A tree traversal, starting from the given $variable$**TYPEa and** leading to the developed concepts of the formed virtual layer, can provide the contribution of a single variable to the development of concepts or the virtual counterpart of the initially given variable. Also, the opposite can be obtained provided the availability of the governing rules while traversing the tree or the availability of the development trace as that given by (Eq-4.36).

The traceability feature described by (Eq-4.36) provides the capability to the developed model to exchange information at a meta-cognitive level. Figure-4.13 provides a schematic for the exchange of information between concepts [4.6].

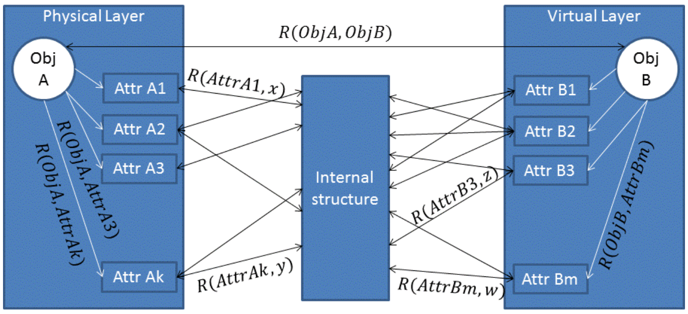


Figure-4.13 References among the occurring and the virtual entities.

Traceability informs about the capability of the model to perform tracking the transformations from input to output while the term trail is required to denote the capability of achieving the meta-cognitive communication without actually parsing the formed tree. For this reason, borrowing the OAR model [4.5] to express the Long-Term Memory (LTM) [4.5] defined as $OAR \triangleq (O, A, R)$ with O to denote the involved objects, A to stand for the objects' attributes, and R representing the developed relations, it provides the capability to make references from the physical level to the virtual level and vice-versa. Figure-4.13 depicts the developed relations and references.

4.5 Constraints

The introduced constraints in the operation of the model are classified into three classes,

1. the medical therapies and cures provided by the medical professionals in the form of protocols.
2. the expressed personal preferences of the supported individual.
3. the conditions of the individual's health status.

In addition, the constraints component is equipped with two handles, one to receive the values and the types of constraints and the other to export the rules defined by the received constraints. The constraints component develops rules produced by the provided medical, personal, and holding restrictions and which are going to be applied on the elements of the processing layer,

$< StaticConstraints\mathbf{PC} = ($

$\quad ConstraintsFlowData($

$\qquad < \mathbf{I} :: (Medical\mathbf{TYPE},\ Preferences\mathbf{TYPE},\ HealthStatus\mathbf{TYPE}) >$

$< \mathbf{O} :: (Constriant\mathbf{TYPE}) >$

$< \mathbf{UDM} ::< RuleBasedSystem\mathbf{ST}, Handle8\mathbf{ST}, Handle9\mathbf{ST} >)\mathbf{PC}$

$) >$ (Eq-4.37)

with

$RuleBasedSystem\mathbf{ST} = \ ($

$($

$< Rules\mathbf{ST}($

$< MedicalProtocol\mathbf{TYPE}|MedicalProtocol \in \mathbf{M} >$

$< SetPreferences\mathbf{TYPE}|SetPreferences \ \in \mathbf{P} >$

$) >$

$\parallel < Facts\mathbf{ST}($

$< MedicalOrders\mathbf{TYPE}|MedicalProtocol \in \mathbf{M} >$

$< CurrentConditions\mathbf{TYPE}|CurrentConditions \ \in (\mathbf{P} \cup \mathbf{M}) >$

$< PersPreferences\mathbf{TYPE}|PersPreferences \ \in \mathbf{P} >$

$) >$

$)$

$< ProductionRules\mathbf{ST} >$

$< Priorities\mathbf{ST} >$

$< Inference\mathbf{ST}($

$< ForewardChaining\mathbf{ST} >$

$\parallel < BackwardsChaining\mathbf{ST} >$

$) >$

$< ConflicResolution\mathbf{ST} >)$ (Eq-4.38)

where $\mathbf{M} = \{MedicalProtocol_1, \ldots . MedicalProtocol_k\}$ the set of available medical protocols, $\mathbf{P} = \left\{PersPreference_1, \ldots . PersPreference_j\right\}$ and $j, k \in \mathbf{N}$**.** The produced inference at $Inference\mathbf{ST}$ element is supplied from $Handle8\mathbf{ST}$ element which, in turn, is supplied by the input section of (Eq-4.37). The formal definition of the internal structure, the attributes, and the developing relations of **M** and **P** sets is beyond the current scope.

4.6 Controlling and Regulation

The self-controlling component functions as a regulator to the operation of the main layer structure of the model. The component considers the layered structure as the controlling process and uses a set of control tools. The static behavior of the self-controlling component is affected by the functioning of two participating handles, the control tools, and the decision making sub-component:

$$
\begin{aligned}
&< SelfControlFlowData\mathbf{PC} = \ (\\
&\quad < \mathbf{I} :: (VirtualData\mathbf{TYPE}) > \\
&\quad < \mathbf{O} :: (OutControlData\mathbf{TYPE}) > \\
&\quad < \mathbf{UDM} ::< ControlInfrastructure\mathbf{ST}, Handle5\mathbf{ST}, Handle6\mathbf{ST} >)\mathbf{PC} \\
&\quad) > \qquad \text{(Eq-4.39)}
\end{aligned}
$$

with

$$
\begin{aligned}
&ControlInfrastructure\mathbf{ST} = \ (\\
&(\\
&\qquad \| < Controller\mathbf{ST} > \\
&\qquad \| < Regulator\mathbf{ST} > \\
&\qquad \| < Observer\mathbf{ST} > \\
&\qquad \| < ControlTools\mathbf{ST}(\\
&\qquad \| < Controllability\mathbf{ST} > \\
&\qquad \| < Observability\mathbf{ST} > \\
&\qquad \| < Maintainability\mathbf{ST} > \\
&\qquad \| < Stabilizability\mathbf{ST} > \\
&\qquad \| < Reachability\mathbf{ST} > \\
&\qquad \| < Detectability\mathbf{ST} > \\
&\qquad \| < Stability\mathbf{ST} > \\
&) \ ) \qquad \text{(Eq-4.40)}
\end{aligned}
$$

Where, a typical structure of a control tool of the $ControlTools\mathbf{ST}$ component must be structured in such a way that it can accommodate matrices of the attributes of the involved concepts at the virtual layer, such as:

$$
\begin{aligned}
&TypicalControlTool\mathbf{ST} = \ (\\
&< SystemVectors\mathbf{ST} = (\\
&\quad (
\end{aligned}
$$

$< Avector\mathbf{ST} >< Bvector\mathbf{ST} >< Cvector\mathbf{ST} >< Dvector\mathbf{ST} >$

$)$

$< XstatesVector\mathbf{ST} >$

$< UinputVector\mathbf{ST} >$

$< YoutputVector\mathbf{ST} >$

$< ToolTest\mathbf{ST} >) >$ (Eq-4.41)

where, $ToolTest\mathbf{ST}$ is the place holding the result of the performed tests of Controllability, Observability, Maintainability, Stabilizability, Reachability, Detectability, and Stability, that cause the activation of the appropriate corresponding corrective actions.

4.7 Knowledge regulation

The knowledge component of the model provides those services to other components that require the participation of knowledge in order to accomplish their specific tasks. The layered structure, the constraints, or the self-controlling components demand for knowledge in order to acquire additional information to perform the proper decisions in order to take the appropriate actions. Each time a component or an element of the layered structure that needs to resolve an issue, it exports queries to the knowledge component. For instance, a query can request the synonyms or the antonyms of a given item or a concept and the knowledge component returns a complete set of concepts with attributes and relationships instantiated at that occurring instance of time. Object-orientation provides the means to implement such demanding data structures consisted of data structures with assigned values and pointers defining their interrelationships. Hence, the knowledge component must present the proper internal infrastructure to satisfy the request posed by the associated components of the model.

Approaching the knowledge requirements of the Home UbiHealth model as a universal structure, i.e., a hyper-structure, $\mathbb{HR}$, as it is defined in [4.6], it is consisted of a set of words. For a given word, the corresponding concept is addressed within the hyper-structure and the rest of the contents of the hyper-structure are classified as sub-concepts, super-concepts, equivalent, opposite, or indifferent. Recalling from [4.5] the $OAR \triangleq (O, A, R)$ model, O for objects, A for attributes, and R for the developed relations, for the memory representation and extending the cross-product or Cartesian product among the objects, then knowledge can be represented as a concept network or objects network defined in [4.6] as $\mathfrak{K} = \triangleq \mathfrak{N}: {}_{i=1}^{n}X c_i \rightarrow {}_{i=1}^{n}X c_i$ with c_i to stand for the representation of concepts as objects and $i \in \mathrm{N}$. The semantic environment, $\Theta,$ is defined [4.6] as correspondences between a non-empty set U of objects and a non-empty set M of attributes and the set R of their interrelationships [4.6] defining $\Theta = (U, M, R)$ or $R: U \rightarrow U | U \rightarrow M | M \rightarrow U | M \rightarrow M.$ Then, the semantic environment can be used to define [9.4] formally the entity of concept as the tuple of $C = (O, A, R^c, R^i, R^o)$ with the first three terms refer to the semantic environment and the rest two terms refer to the input and the output relations [4.6].

There are three significant properties that arise from the given definition of concepts,

1. the intension which is captured as a set of additional pairs of attributes-relations,
2. the extension which can take advantage of the objects' inheritance property instantiating constructs of objects,
3. the autonomy which is captured with its internal capability to interrelate and create additional forms of knowledge.

In order to perform algebraic manipulations on concepts it is required the framework in which to perform the adequate operations. Concept algebra provides such a framework consisted of three terms: first, the concepts that participate in the manipulations, second, the operations to be performed on the concepts, and third, the environment within which the operations must apply on defined concepts. Hence, the relation $Concept\ Algebra = CA \triangleq (C, OP, \Theta) = (\{O, A, R^c, R^i, R^o\}, \{\cdot_r, \cdot_c\}, \Theta)$ presented in [4.6] provides the means to perform relational $\cdot_r$ and compositional $\cdot_c$ manipulations on determined concepts.

The knowledge component of the model must be capable of accommodating the necessary structural elements that allow the communication of this component with the rest of the model's components. Hence, $handle12\mathbf{ST}$ must be accommodated within the knowledge component. In addition, the knowledge component must present a structure to accommodate the objects that represent concepts. Therefore, every object must be uniquely characterized, it must contain a set

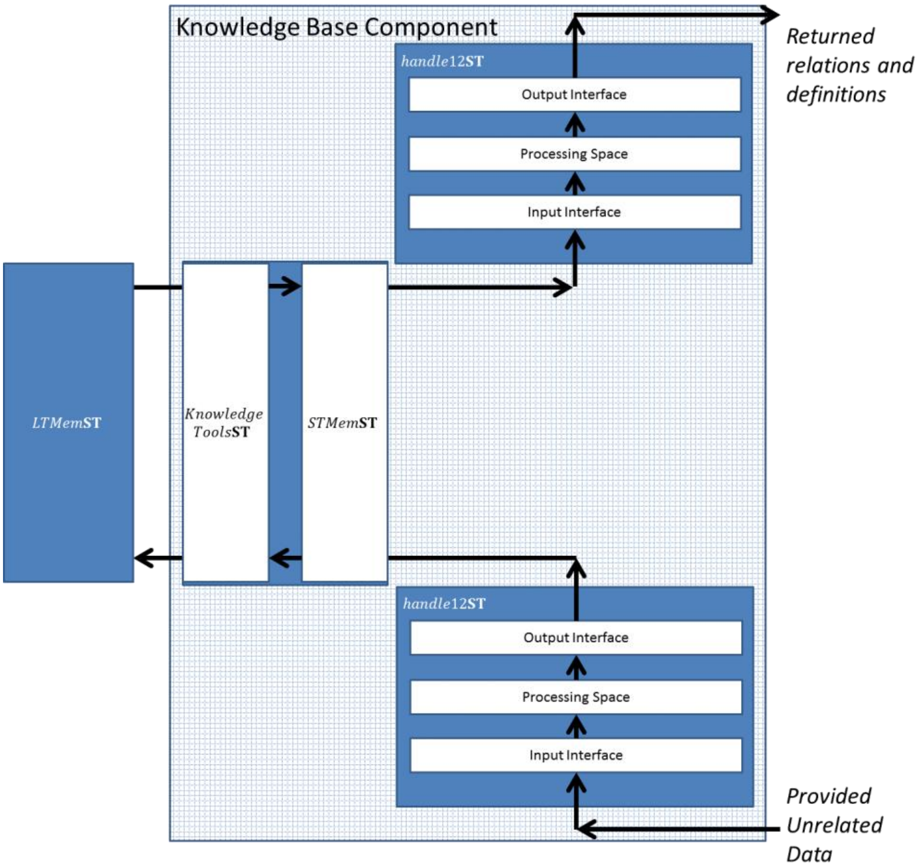


Figure-4.14 Building blocks of the knowledge component of the model.

of attributes, a set of other objects included in the integration of the compound object, it must include the internal among the contained objects, it must contain the external input and the external output relations with other object in addition to the already included objects, it must discretely contain references to related synonyms and antonyms [4.6]. The representation of a conceptual element is described formally by the relation of (Eq-4.42). The element of (Eq-4.42) is used by the structure of the knowledge component of the model. The structure of the knowledge component must contain a structure that can use and administer the concepts from the supporting memory facilitated by processes that can perform the designed operations. A structure analogous to a tree is required to represent the concepts (Eq-4.42) in the residing memory:

$$Concept\mathbf{ST} = \ ($$

$$< CID\text{: }\mathbf{TYPE}|CID = \{C_1, C_2, \ldots, C_n\} \wedge (0 \leq CID \leq FFFF\mathbf{TYPE}) >$$

$$< CAttribute\text{: }\mathbf{TYPE}|CAttribute\mathbf{TYPE} = \{A_1, \boldsymbol{A_2}, \ldots, \boldsymbol{A_m}\} >$$

$$< CObject\text{: }\mathbf{TYPE}|CObject\mathbf{TYPE} = \{O_1, \boldsymbol{O_2}, \ldots, \boldsymbol{O_k}\} >$$

$$< CIntRel\text{: }\mathbf{TYPE}|CIntRel\mathbf{TYPE} = \{A_m \mathbf{x}\ \boldsymbol{O_k}\} \wedge \boldsymbol{k}, \boldsymbol{m} \in \boldsymbol{N} >$$

$< CExtInput\text{: }\mathbf{TYPE}|CExtInput\mathbf{TYPE} = \{CID\mathbf{TYPE}\ \mathbf{x}\ CID\mathbf{TYPE}\} >$

$< CExtOutput\text{: }\mathbf{TYPE}|CExtOutput\mathbf{TYPE} = \{CID\mathbf{TYPE}\ \mathbf{x}\ CID\mathbf{TYPE}\} >$

$< CSynonyms\text{: }\mathbf{TYPE}|Equivalence\left(C_p\mathbf{TYPE}, C_q\mathbf{TYPE}\right) = T\mathbf{B} \wedge \boldsymbol{p}, \boldsymbol{q} \in \boldsymbol{N} >$

$< CAntonyms\text{: }\mathbf{TYPE}|Equivalence\left(C_p\mathbf{TYPE}, C_q\mathbf{TYPE}\right) = F\mathbf{B} \wedge \boldsymbol{p}, \boldsymbol{q} \in \boldsymbol{N} >$

$< CTimeStamp\text{: }\mathbf{Time}|0000{:}\,00{:}\,00{:}\,00{:}\,00{:}\,00 \leq CTimeStamp\mathbf{Time} \leq$
$9999{:}\,12{:}\,31{:}\,24{:}\,59{:}\,59 >)$ (Eq-4.42)

The knowledge component communicates with the rest of the model's components through the $handle12\mathbf{ST}$ element. The received queries are acquired by $handle12\mathbf{ST}$ and then placed in the short term memory bank. Then, each query is processed using the long term memory with the available tools. The produced results are placed back in the short term memory and from there; the results are placed back to the $handle12\mathbf{ST}$ element in order to return the results of the query to the requesting component. The static behavior of the knowledge component is described by (Eq-4.43) giving a detailed description of the constituting building blocks:

$< KnowledgeFlowData\mathbf{PC} = ($

$< \mathbf{I} :: (QueryData\mathbf{TYPE}) >$

$< \mathbf{O} :: (QueryData\mathbf{TYPE}) >$

$< \mathbf{UDM} ::< handle12\mathbf{ST}, STMem\mathbf{ST}, LTMem\mathbf{ST}, KnowledgeTools\mathbf{ST} >$

$)\mathbf{PC} >$ (Eq-4.43)

The building blocks of the internal structure of the knowledge component is presented in Figure-4.14 with the among them developing relations.

4.8 Dynamic Behavior: Events and Interrupt Handling

The functional behavior of the Home UbiHealth model is characterized by the operation of the involved processes. A process is initiated at the input of the Event Driver component with the acquisition of raw data and signals from the physical environment and it is continuously evolving over time until the incoming data and signals vanishes from the input. In other words, a process P is continuously evolving in time t by changing its contents according to the changes $f: P \rightarrow t$ of its input over time or $P = f(t)$. From a discrete space view, the process P can take any value from the available set of values defined in the set of $P = \{p_1, p_2, \dots, p_n\}\ with\ n \in N$. The changes in the input of the process are realized by two main ways:

1. By the interrupts directed from the occurring events in the physical environment, and
2. By the semantic interrupts raised from the interaction among the values of acquired data

The semantic interrupts are raised by the application of medical and personal preferences, which place additional barrier and limitations to the evolvement of the process. The interrupts intervene in the flow of the process' operations causing the flow of alternative paths and the request to change values of some of the variables in the physical environment. For instance, the change in the infusion rate of an implanted nano-pump can be caused by the evaluation of the values of a set of variables which are not associated physiologically but directly associated semantically performing a diagnosis. In such cases, the process must change some of its constituting variables which is accomplished by sending the proper output at the ports of the Event Driver to be transferred through the interfaces to the dispersed devices in the ubiquitous computing environment.

The process evolvement must always consider the changes occurring in the physical environment, which is accomplished by the dispersed sensors within the home environment. In this direction, the individual at home must be continuously supported even when moving within the house and sensed by different sets of devices providing support to the individual's mobility. The individual is interacting with a set of stationary, mobile, and implanted devices with the contents of the set to change as the user is moving within the house. Such a behavior, at any instant of time, is causing the support of the individual by a different set of devices but the supporting applications remain exactly the same. In order to support the freely moving individual by the same set of software applications, the software applications must be capable of performing migration by transferring the application's control from a device to another, including a proactive anticipation.

The decisions taken by the process must be limited by the availability of the placed medical constraints and the personal preferences on the current status of the individual's health. The judgments performed by the evolving process are supported by the existing state of knowledge, which can be expressed in various forms and at different levels. The availability of knowledge in the process evolvement provides the capability to make the proper decisions, whenever required, in a knowledgeable manner providing operational and functional autonomy to the carrying processes.

The availability of knowledge in the process evolvement provides the capability to access the process at different levels of abstraction. A process, at any instant, can be described with a different set of parameters depending upon the available values that uniquely identify the process and its state. The available values of the parameters imply a more detailed, at a lower level, abstraction while similarly, a less detailed, at a higher level, abstraction. In other words, taking advantage of the process abstract representation, it can be accessed with different levels of abstraction depending upon the available, each time, values of the set of parameters. For instance, a medical professional can interact with a process providing a very detailed set of parameters related to glucose level in the individual’s blood while a supporting family member can be limited by the use of a single term referring to the concept of diabetes. Hence, the model provides the capability to interact at different levels of abstraction provided the availability of the set of the identifying, each time, parameters of the corresponding concept.

The Home UbiHealth model uses a single process that can support the needs of the health status of the individual at home. The employed process must be capable of satisfying the occurring simultaneous and concurrent needs of the supported individual. The process has a single input and a single output ports at the Event Driver component. The input can occur at any time when an event takes place in the individual’s environment or when the formed circumstances require an action to support the individual’s health status. Hence, events and interrupt handling mechanism must administer the incoming requests at the Events Driver component described by (Eq-4.44). The used symbols of ↯ is used to denote an interrupt occurring when the symbol’s @ following parameters values have been met and then a dispatching takes place symbolized with ↳ character and the execution of a process is carried using the → symbol while an interrupt’s code starts with the symbol ↗ and ends after the symbol ↗ at the point of execution denoted by the ⊙ symbol.

$HomeUbiHealth\S.InterruptDispatch\mathbf{PC} = HomeUbiHealth\S \rightarrow$

{

$\quad \text{↯}\ @SystemInitialization\mathbf{ST}$

$\quad \text{↳}\ SystemInitialize\mathbf{PC} \rightarrow {}_{SystemStatus\mathbf{BL=}}{}^{\boldsymbol{F}}_{\mathbf{T}}\boldsymbol{R}(\text{↳}\ Scheduler\mathbf{PC} \rightarrow \boxtimes)$

$\quad \text{↯}\ @MedicalProfessionalInterrupt\odot$

$\nearrow (\Lsh\ Constraints\mathbf{PC} \rightarrow @UserInterrupt \coloneqq off\ ) \searrow \odot$

$ϟ\ @UserInterrupt\odot$

$\nearrow (\Lsh\ LayeredStructure\mathbf{PC} \rightarrow @UserInterrupt \coloneqq off\ ) \searrow \odot$

$ϟ\ @DeviceInterrupt\odot$

$\nearrow (\Lsh\ LayerdStructure\mathbf{PC}\ \rightarrow @DeviceInterrupt \coloneqq off\ )\ \searrow \odot$

$ϟ\ @SemanticInterrupt\odot$

$\nearrow (\Lsh\ LayeredStructure\mathbf{PC}\ \rightarrow @SemanticInterrupt \coloneqq off\ ) \searrow \odot$

$ϟ\ @PeriodicInterrupt\odot$

$\nearrow (\Lsh\ EventDriver\mathbf{PC}\ \rightarrow @DeviceInterrupt \coloneqq off\ ) \searrow \odot$

$ϟ\ @ClockInterrupt\mathbf{ST}$

$\nearrow (\Lsh\ EventDriver\mathbf{PC}\ \rightarrow @DeviceInterrupt \coloneqq off\ ) \searrow \odot$

$\} \rightarrow HomeUbiHealth§$ (Eq-4.44)

The realized interrupts in (Eq-4.44) must be prioritized for the proper operating of the model supporting the individual's health status. The model's internal operations are assigned with the lowest priorities while the highest priorities are privileged to the participating human roles. The model is featured with seven (7) priority levels going from the lower to the higher priority level:

$HomeUbiHealth§.PrioritiesAllocation\mathbf{PC} =\ HomeUbiHealth§ \rightarrow$

$\{$

$InitializeSystem\mathbf{PC}$

$\| AdvanceClock\mathbf{PC}$

$\| PrioritizePeriodicEvents\mathbf{PC}$

$\| PrioritizeDevices\mathbf{PC}$

$\| PrioritizeUser\mathbf{PC}$

$\| PrioritizeSemantics\mathbf{PC}$

$\| PrioritizeMedicalProfessional\mathbf{PC}$

$\} \rightarrow HomeUbiHealth§$ (Eq-4.45)

The presence of an interrupt with the correspondingly assigned priority directly affects the status of the Event Driver component receiving the realized signals. In particular, the occurring interrupts place their requests at sub-component $handle0\mathbf{ST}$ whose changed contents allow the process $EventDriver\mathbf{PC}$ to cause changes to the contents of $handle1\mathbf{ST}, handle6\mathbf{ST},$ and $handle7\mathbf{ST}$ too. Those changes at the

Event Driver’s handles are propagating through the handles to the rest of the handles within the system of the model.

From Figure-4.12 and for reasons of simplicity it can be stated that $handle1\mathbf{ST} = handle2\mathbf{ST} = handleA\mathbf{ST}.$ Similarly, it can be stated that $handle3\mathbf{ST} = handle4\mathbf{ST} = handleB\mathbf{ST}$ and in the same fashion, $handle5\mathbf{ST} = handle6\mathbf{ST} = handleC\mathbf{ST},$ while the next pair of $handle7\mathbf{ST} = handle8\mathbf{ST} = handleD\mathbf{ST},$ and $handle9\mathbf{ST} = handle10\mathbf{ST} = handleE\mathbf{ST},$ and $handle11\mathbf{ST} = handle12\mathbf{ST} = handleF\mathbf{ST}.$ With those equivalences between the pairs of the involved handles, the spreading of the realized interrupt values in the discrete components of the model becomes more intuitive:

$$
\begin{aligned}
&HomeUbiHealth\S.PrioritiesAllocation\mathbf{PC} = \ HomeUbiHealth\S \rightarrow \{\\
&\| EventDriver\mathbf{PC} = (\\
&\qquad < I :: (@interrupt\mathbf{TYPE}) >\\
&\qquad < O :: (handle0\mathbf{ST}, handleA\mathbf{ST}, handleD\mathbf{ST}) >\\
&\qquad < \mathbf{UDM} :: (EventDriverInternalStructure\mathbf{ST} >)\mathbf{PC}\\
&\| LayeredStructure\mathbf{PC} = (\\
&\qquad < I :: (handleA\mathbf{ST}) > < O :: (handleB\mathbf{ST}, handleE\mathbf{ST}, handleF\mathbf{ST}) >\\
&\qquad < \mathbf{UDM} :: (LayeredStructure\mathbf{ST} >)\mathbf{PC}\\
&\| SelfControl\mathbf{PC} = (\\
&\qquad < I :: (handleB\mathbf{ST}) > < O :: (handleC\mathbf{ST}) >\\
&\qquad < \mathbf{UDM} :: (SelfControl\mathbf{ST} >)\mathbf{PC}\\
&\| Constraints\mathbf{PC} = (\\
&\qquad < I :: (handleD\mathbf{ST}) > < O :: (handleE\mathbf{ST}) >\\
&\qquad < \mathbf{UDM} :: (onstraints\mathbf{ST} >)\mathbf{PC}\\
&\| Knowledge\mathbf{PC} = (\\
&\qquad < I :: (handleF\mathbf{ST}) > < O :: (handleF\mathbf{ST}) >\\
&\qquad < \mathbf{UDM} :: (Knowledge\mathbf{ST} >)\mathbf{PC}\\
&\} \rightarrow HomeUbiHealth\S \qquad\qquad \text{(Eq-4.46)}
\end{aligned}
$$

4.9 Performing Applications Migration

There are case situations in the Home UbiHealth model’s environment where it is required a software application to leave from the device it is currently operating and be transferred to another one. The computing environment of the device that receives the migrating application can be completely different than the device the application is migrating from, it can be similar, or it can be exactly the same. In such cases, the model realizes the proper migration policy to continuously support the health status.

4.9.1 Pre-conditions

Recalling the procedure $InterruptDispatch\mathbf{PC}$ of (Eq-4.44) that activates, through the raised $@UserInterrupt,$ the $LayeredStructure\mathbf{PC}$ component must resolve and classify the kind of interrupt to act accordingly. The same component $LayeredStructure\mathbf{PC}$ is activated by the $@SemanticInterrupt$ due to the formed situations at the developing context. Also, in the same fashion, a device, $@DeviceInt,$ or a software application $@ApplicationInt,$ can raise an interrupt that results in the activation of $LayeredStructure\mathbf{PC}$ that initiates, synchronizes, and coordinates the migration of the running application to another available computing environment, if it is capable, by the raise $@DeviceInterrupt$ of the $InterruptDispatch\mathbf{PC}$ component:

$HomeUbiHealth\S.LayeredStructure\mathbf{PC} = \ HomeUbiHealth\S \rightarrow \{$

$\|(\lightning\ @UserMovementInt\odot$

$\nearrow (↳\ ProcessMigrationSupport\mathbf{PC} \rightarrow @UserMovementInt\odot := off)$

$\searrow \odot)$

$\|(\lightning\ @ContextualSituationInt$

$\nearrow (↳\quad \{$

$\blacklozenge ContextualConditions\mathbf{N} =$

$|1 \rightarrow \oiint_{iN=1}^{nN} R@DeductiveInference_i\mathbf{ST}$

$↳\ ProcessMigrationSupport_i\mathbf{PC}$

$|2 \rightarrow \oiint_{jN=1}^{nN} R@InductiveInference_j\mathbf{ST}$

$↳\ ProcessMigrationSupport_j\mathbf{PC}$

$|3 \rightarrow \oiint_{kN=1}^{nN} R@AbductiveInference_k\mathbf{ST}$

$↳\ ProcessMigrationSupport_k\mathbf{PC}$

$|4 \rightarrow \oiint_{lN=1}^{nN} R@AnalogyInference_l\mathbf{ST}\ ↳\ ProcessMigrationSupport_l\mathbf{PC}$

$|\ \blacklozenge\sim\ \rightarrow \varnothing$

$\}\ \rightarrow @ContextualSituationInt\odot := off)\searrow \odot)$

$\|(\lightning\ @DeviceInt\odot$

$\nearrow (\Rsh ProcessMigrationSupport\mathbf{PC} \rightarrow @DeviceInt\odot := off) \searrow \odot)$

$\|(\lightning\ @ApplicationInt\odot$

$\nearrow (\Rsh ProcessMigrationSupport\mathbf{PC} \rightarrow @ApplicationInt\odot := off) \searrow \odot)$

$\|(LayersProcessing\mathbf{PC})$

$\} \rightarrow HomeUbiHealth\S$ (Eq-4.47)

4.9.2 Polymorphism and types of Migration

The migration of a running software application on one of the dispersed devices of the ubiquitous computing environment is based on the comparison of the semantic equivalency and coincidence of the departing and the arriving computing environment in alignment with the evolving context. In other words, the software application is going to continue its operation at the migrated device from the point it was interrupted at the migrating device providing exactly the same operational and functional behavior. Such required coincidence can be provided by the escorting taxonomies of the knowledge component that communicate with layered structure through $handle11\mathbf{ST} = handle12\mathbf{ST} = handleF\mathbf{ST}.$ The carried migration procedure is formally described by (Eq-4.48) providing the set of types of the input and the output parameters along with the type of the operating structure.

The dynamic operation of the $ProcessMigrationSupport\mathbf{PC}$ procedure includes the examination of the migrated and migrating devices at the ACD level using the evaluation ♦ operator on the participating structure of $fromACD(timing)$ and $toACD(timing)$. Considering that the application can be decomposed into segments and examining the availability and appropriateness of each migrating segment, then depending upon the resulting comparison that the migrating application can be replaced if the comparison returns 1, adapted if the comparison returns 2, or replaced and adapted if the returning result of the comparison is equal to 3, running the corresponding appropriate procedure:

$ProcessMigrationSupport\mathbf{PC}($

$< I :: \ fromACD(timing)\mathbf{ST}, toACD(timing)\mathbf{ST},$

$timing\mathbf{TM}, InSemaphor\mathbf{BL} >$

$< \mathbf{O} \ :: outACD(timing)\mathbf{ST}, OutSemaphor\mathbf{BL} >$

$< UDM :: MigratingApplication\mathbf{ST} > \triangleq$

$\{$

$$
\begin{aligned}
&\rightarrow (\blacklozenge(fromACD(timing) == toACD(timing))\\
&\rightarrow (testACD(timing) = fromACD(timing))\\
&\quad |(\blacklozenge\sim) \rightarrow\\
&\{\\
&\rightarrow SemanticEquivalence\boldsymbol{PC}\\
&\rightarrow (testACD(timing) = equivalentACD(timing))\\
&\}\\
&\rightarrow (_{iN=1}^{n}\mathbf{R}\\
&\blacklozenge(inACD(timing)\mathbf{ST}.\mathrm{ApplicationPart}_i\\
&\qquad\qquad == toACD(timing)\mathbf{ST}.\mathrm{ApplicationPart}_i) =\\
&|1 \rightarrow Replace\boldsymbol{PC} < \boldsymbol{I} :: testACD(timing), InSemaphor\boldsymbol{BL} >< \boldsymbol{O}\\
&\qquad\qquad :: toACD(timing), OutSemaphor\boldsymbol{BL} >< \boldsymbol{UDM}\\
&\qquad\qquad :: ReplaceCodeApp\boldsymbol{PC} >\\
&|2 \rightarrow Adapt\boldsymbol{PC} < \boldsymbol{I} :: testACD(timing), InSemaphor\boldsymbol{BL} >< \boldsymbol{O}\\
&\qquad\qquad :: toACD(timing), OutSemaphor\boldsymbol{BL} >< \boldsymbol{UDM}\\
&\qquad\qquad :: ReplaceParametersApp\boldsymbol{PC} >\\
&|3 \rightarrow Replace\&Adapt\boldsymbol{PC} < \boldsymbol{I} :: testACD(timing), InSemaphor\boldsymbol{BL} >< \boldsymbol{O}\\
&\qquad\qquad :: toACD(timing), OutSemaphor\boldsymbol{BL} >< \boldsymbol{UDM}\\
&\qquad\qquad :: ReplaceCodeAndAdaptParametersApp\boldsymbol{PC} >\\
&|\ \blacklozenge\sim\ \rightarrow \varnothing\\
&)\,\} \qquad\qquad \text{(Eq-4.48)}
\end{aligned}
$$

The property of polymorphism is accomplished by achieving to maintain the provision of the same functionality while changing the content and the internal structure of the migrating software application. The migration requirements dictate the type of migration that are limited to,

- the replacement of the migrating application by a semantically equivalent piece of code
- the adaptation of the parameters of the migrating software application to meet the demands of the new hosting device
- the enforcement of both replacement and adaptation too.

Polymorphism is a property of the system that allows to preserve the same functionality even when the hosting computer environment is different. Then, the types of migration can be three in the model and they can be applied by (Eq-4.48) based on the semantic equivalence of the migrating and migrated environments.

4.10 Autonomy

The operation and the behavior of the model must ensure that it achieves the purposes and the anticipated goals. At each event, internal or external, the system must make the proper decisions and act accordingly. The capability of the model to make decisions upon the received signals and data, which can support with the adequately designed actions that keep the system within the frame of its goals is considered to be the property of autonomy. In the Home UbiHealth model the designed infrastructure of software structures and procedures are capable of carrying out such processes that the patients' health status is supported properly at all times. The software infrastructure must present those mechanisms that realize the individual's mobility within the house, moving from one location to another, while it is prepared to perform a series of actions that ensure the continuous support of the individual at all locations by allowing the transfer of the supporting applications at the devices near the location of the individual. This is accomplished by taking the proper decisions based on evaluations of the received data and signals. Hence, autonomy is the property related with the intrinsic capability of the system to perform decisions on received data and take the proper actions that allow the system to achieve its purposes.

The dynamic behavior of the model B is determined by the functionality of the autonomic intellectual responsiveness [4.7] of the system, which depends on a set of determining parameters [4.8] (Eq-4.49). The dynamically changing description of the system's responsiveness is determined by the composition of the set

$$B = \left\{B_t, B_e, B_{int}, B_g, B_d, B_p, B_{inf}\right\} \qquad \text{(Eq-4.49)}$$

involving the following concepts in the determination of the system's behaviors as they are described in [4.13]

B_t: concept of time

$$B_t \triangleq {}_{i=1}^{n}R(@time_i\mathbf{TM} \Lsh_{time} Process_i \qquad \text{(Eq-4.49a)}$$

B_e: involved events,

$$B_e \triangleq {}_{i=1}^{\;n}R(@event_i\mathbf{TYPE} \text{↳}_{event}\ Process_i \qquad \text{(Eq-4.49b)}$$

B_{int}: applied instructional interrupts

$$B_{int} \triangleq {}_{i=1}^{\;n}R(@int_i \odot\text{↳}_{int}\ Process_i \qquad \text{(Eq-4.49c)}$$

B_g: set goals for the system

$$B_g \triangleq {}_{i=1}^{\;n}R(@goal_i\mathbf{TYPE} \text{↳}_{goal}\ Process_i \qquad \text{(Eq-4.49d)}$$

B_d: used set of decisions

$$B_d \triangleq {}_{i=1}^{\;n}R(@decision_i\mathbf{TYPE} \text{↳}_{decision}\ Process_i \qquad \text{(Eq-4.49e)}$$

B_p: developed perception

$$B_p \triangleq {}_{i=1}^{\;n}R(@perception_i\mathbf{PC} \text{↳}_{perception}\ Process_i \qquad \text{(Eq-4.49f)}$$

B_{inf}: way of performing inferences

$$B_{inf} \triangleq {}_{i=1}^{\;n}R(@inference_i\mathbf{TYPE} \text{↳}_{inference}\ Process_i \qquad \text{(Eq-4.49g)}$$

The autonomy in the Home UbiHealth model is directly related to the capability of fragmenting the software applications into parametrized modules. In addition the semantic equivalence of the applications’ fragments can be evaluated at run-time. Hence, in a migrating application some of the fragments of the application can be replaced and then be adjusted in order to meet the migration demands. In order to achieve the semantic equivalence in a migration in the model, the status of the migrating and the status of the migrated contexts must be evaluated in order to find the required replacements and adjustments. Resolving the contextual fit of a migrating application to the migrated context’s demands is an operation that requires the capability to make knowledgeable decisions. The decisions can be implemented synthesizing the available applications fragments to achieve a semantical matching with the demands of the migrated computing environment while keeping exactly the same functionality for the migrated software application.

4.11 Interacting using Concepts

The interaction with the Home UbiHealth model using concepts requires the availability of a properly formed interface for the potential users of the system. An interface can provide the means to either provide the input / output of an identified concept or receive as input of compound concepts consisted of the aggregation of many concepts and returns discretely identified concepts

In (Eq-4.42), which provides the definition of a concept, it is shown that concept is a composite structure that includes identification of the concept, the attributes, the holding internal and external relationships, and time. Also, the provided definition is drawn from the mentioned OAR model of the universal context and the 5-tuple definition of an abstract concept: $C = \{O, A, R^c, R^i, R^o\}$. In other words, an abstractly defined concept can be instantiated with specific values at a given time instance. A compact version of (Eq-9.14) is given with elements of type $\mathbf{TYPE}$ as

$$Concept\mathbf{ST} = (< CID{:}\, \mathbf{TYPE} >< CAttribute{:}\, \mathbf{TYPE} >< CObject{:}\, \mathbf{TYPE} >< CIntRel{:}\, \mathbf{TYPE} >< CExtInput{:}\, \mathbf{TYPE} >< CExtOutput{:}\, \mathbf{TYPE} >< CSynonyms{:}\, \mathbf{TYPE} >< CAntonyms{:}\, \mathbf{TYPE} >< CTimeStamp{:}\, \mathbf{Time} >) \qquad \text{(Eq-4.50)}$$

The above relation reveals that concepts instantiation is directly related to time and the constituting elements can be expressed as time-varying parameters providing a dynamic nature to the structure. In other words, concepts are interrelated and dynamically evolving functions of time.

4.11.1 Concept Algebra for users interactions

The definition of a set of operations that can be performed on defined concepts [4.10] provides the capability to develop composite concepts. The provided operations in [4.10] can lead to the development of interrelated concepts forming hierarchical structures of concepts evolving in time. The proposed operations on concepts can be classified into relational and compositional operations.

The relational operations include:

- related concepts, $\leftrightarrow$, providing the intersection of the common characteristics of any two given concepts.

- independent concepts, $\nleftrightarrow$, providing the intersection of the semantic characteristics of any two given concepts.
- sub-concept relation, $\prec$, providing the subset of the characteristics of a concept drawn from the characteristic of another concept.
- super-concept relation, $\succ$, providing the subset of the characteristics of a concept included in the characteristic of another concept.
- synonym operation, $=$, providing the complete correspondence among the characteristics of any given pair of concepts.
- antonym operation, $!=$, revealing the opposition or indifference among the characteristics of any given pair of concepts.
- consistence operation, $\cong$, providing the common characteristics of a concept as a subset of the characteristic of another concept.
- comparison operation $\sim$, providing the difference in the cardinality of a given pair of concepts.
- definition operation, $\triangleq$, providing the specific determination of the characteristics of a concept or pairs of concepts.

The compositional operations on concepts include:

- the operation of inheritance, $\Rightarrow$, allowing a concept to inherit the characteristics of another.
- extension, $+\Rightarrow$, receiving additional characteristics.
- tailoring, $-\Rightarrow$, adapting the characteristics to specific values.
- substitution, $\sim\Rightarrow$, allowing the replacement of characteristics.
- composition, $\uplus$, developing composite concepts.
- decomposition, $\pitchfork$, considering the constituting concepts.
- aggregation/generalization, $\Lleftarrow$, augmenting and generalizing the characteristics.
- specification, $\vdash$, providing the characteristics' values.
- instantiation, $\longmapsto$, providing a snapshot of a concept's state.

Concept Algebra is necessary to perform the manipulations among the defined concepts administered by the model and used by the interacting users. The medical professionals, the supporting nursing personnel, the family members and the supported individual must be capable to interact with the model's infrastructure using high level conceptual constructs. The development of such an interface is necessary in

1) Let the user place a query on the model's resources about the status of a given concept for a given instant of time
2) Receive information about the status of some concepts

A properly designed monitoring software application must facilitate the user's interactions providing proper guidance with the messages driving the user's actions. The core functionality of such a module is the reception of queries, the examination, the analysis, the decomposition, the testing, and then, the provision of the tests' results. The decomposition of the received queries must be performed by the decomposition operation that involves,

$$C(query) = c_{sconcept}(O_s, A_s, R_s^c, R_s^o, R_s^o) \;{}_{j=1}^{cardin|c|}\pitchfork\; c_j(O_j, A_j, R_j^c, R_j^o, R_j^o) \quad \text{(Eq-4.51)}$$

where, given a concept $C(query)$by the user as a query, the model decomposes with the $\pitchfork$ operator the provided, probably composite, concept $c_j(O_j, A_j, R_j^c, R_j^o, R_j^o)$ against the models known superconcept $c_{sconcept}(O_s, A_s, R_s^c, R_s^o, R_s^o)$ for all the known internal hierarchical structure of concepts performing the operation from $j = 1$ up to the cardinality of the set of all known concepts.

The design aims at the provision of the current status of the occurring and interacting concepts of the holding situation at the virtual layer of the structured layer of the model including the entire set of attributes, properties, and relations within the interacting concepts. The set of concepts occurring at the virtual level of the layer structure of the model must perform the operation of specification in order to provide at the designed interface the set of co-existing sub-concepts in a manner familiar to the users property. Hence, the medical professionals require different information than the family members. Thus, the applied operation of specification provides,

$$C_{user_k}(output) =$$

$$= c_{sconcept}(O_s, A_s, R_s^c, R_s^o, R_s^o)^{cardin|c|}{}_{j=1}R \; [\vdash \; c_j\big(O_j, A_j, R_j^c, R_j^o, R_j^o\big)] \quad \text{(Eq-4.52)}$$

where, the output is processed according to each user's role having C_{user_k} processed outputs, the specification $\vdash$ operation is repeated for all occurring concepts at the virtual level of the layered structure of the model.

4.11.2 Knowledge representation

The availability of Concept Algebra [4.10] provides a formal framework to administer the involved concepts at the virtual layer of the designed layered structure. The existence of formal expressions for concepts provides the opportunity to represent knowledge along with the assistance of the associated ontology. Thus, the involved concepts can participate in the manipulations for the knowledge representation performing the following procedures [4.11]:

$$ConceptInitialization\mathbf{PC} = (< \mathbf{I} :: c_i\mathbf{TYPE} >< \mathbf{O} :: c_i\mathbf{TYPE} >< \mathbf{UDM} :: c_{iOAR\ model}\mathbf{TYPE} > \quad \text{(Eq-4.53a)}$$

$$ConceptComparison\mathbf{PC} = (< \mathbf{I} :: c_i\mathbf{TYPE}, c_j\mathbf{TYPE} >< \mathbf{O} :: result\mathbf{TYPE} >< \mathbf{UDM} :: c_{iOAR\ model}\mathbf{TYPE}, c_{jOAR\ model}\mathbf{TYPE} > \quad \text{(Eq-4.53b)}$$

$$ConceptAttributesExtraction\mathbf{PC} = (< \mathbf{I} :: c_i\mathbf{TYPE} >< \mathbf{O} :: c_{iOAR\ model}\mathbf{TYPE} > < \mathbf{UDM} :: c_{iOAR\ model}\mathbf{TYPE} > \quad \text{(Eq-4.53c)}$$

$$ConceptRelationsAnalysis\mathbf{PC} = (< \mathbf{I} :: c_i\mathbf{TYPE} >< \mathbf{O} :: c_{iOAR\ model}\mathbf{TYPE} > < \mathbf{UDM} :: c_{iOAR\ model}\mathbf{TYPE} > \quad \text{(Eq-4.53d)}$$

$$ConceptDevelopment\mathbf{PC} = (< I :: c_i\mathbf{TYPE} >< \mathbf{O} :: c_j\mathbf{TYPE} >< \mathbf{UDM} :: c_{iOAR\ model}\mathbf{TYPE}, conceptstack\mathbf{TYPE} > \quad \text{(Eq-4.53e)}$$

The procedure administering the conceptual operations occurring at the virtual layer of the model’s layered structure has available the tools given in (Eq-4.53a) through (Eq-4.53e) to manipulate the participating concepts.

4.11.3 Autonomous Conceptual Control

The occurring concepts along with their holding interrelationships feed the Self-Controlling component of the model’s structure. The holding relations feed the static infrastructure of the component activating the evaluation of the terms of a single-order loop control system, which is supported by a set of controlling tools. The applied control applies with the following parameters:

- System: it is represented by the state transition $\bar{A}$ vector of $n -$ dimension formed by the relations among the coexisting concepts c_i **x** c_i of size n **x** n the virtual layer of the model
- State variables: they are represented by the $\bar{X}$ vector of $n -$ dimension
- Input: it is represented by the $\bar{U}$ vector of $m -$ dimension formed with the discrete values of the attributes of the participating concepts denoted in the general form as $c_i = (O_i, A_i, R_i^c, R_i^o, R_i^o)$
- Output: is represented by the $\bar{Y}$ vector of $p -$ dimension values received from the physical environment

Then, the system state space is provided by the set of simultaneous equations given at (Eq-3.32) with the first equation to provide the rate of change in the state variables while the second one provides the (steady-state) output. With the determination of the matrices (or vectors) of $\bar{A}, \bar{B}, \bar{C}, and\ \bar{D},$ the set of control tools (Eq-3.37) through (Eq-3.40) provide the required information for the Decision Making element of the Self-Controlling component. Therefore, the concepts participating at the virtual layer of the layered structure of the model can constitute a system with the concepts to represent the state variables and the concepts receive values for their attributes and their properties from the physical environment, which are directed to the Self-Controlling component to make decisions that actuate the devices and the software applications in the physical level. The required feedback is achieved through the Event Driver component to the layered structure where the virtual layers reside.

The decision making element of the Self-Controlling component can evaluate the control evidences provided by the control-tools by using an inference engine. Using the formally defined components of an inference engine given in [4.12], the decision making must include a procedure testing the availability of evidences,

$TestingOutputAttributes\mathbf{PC} = \{$

$\blacklozenge ExportAttributes\mathbf{N} =$

$|(Precondition\mathbf{B} \wedge Rule\mathbf{B})$

$\rightarrow \oiint {}_{iN=1}^{nN} R@DeductiveInference_i\mathbf{ST}$

$\Lsh ConsequenceOutputAttributes_i\mathbf{PC}$

$|(\mathrm{Rule}\mathbf{B} \wedge Conclusion\mathbf{B})$

$\rightarrow \oiint {}_{jN=1}^{nN} R@InductiveInference_j\mathbf{ST}$

$\Lsh FindAndApplyRuleOutputAttributes_j\mathbf{PC}$

$$
\begin{aligned}
&|(Precondition\mathbf{B} \land Conclusion\mathbf{B}) \\
&\to \oiint_{kN=1}^{nN} R @AbductiveInference_k\mathbf{ST} \\
&\qquad\qquad \Lsh MostProbableOutputAttributes_k\mathbf{PC} \\
&|(Precondition\mathbf{B} \land Rule\mathbf{B} \land Conclusion\mathbf{B}) \\
&\to \oiint_{lN=1}^{nN} R @AnalogyInference_l\mathbf{ST} \\
&\qquad\qquad \Lsh ProportionalOutputAttributes_l\mathbf{PC} \\
&|\ \blacklozenge\sim\ \to \varnothing\ \}
\end{aligned}
\qquad \text{(Eq-4.54)}
$$

According to the definition of a cognitive computer provided by [4.9], a cognitive computer requires the functioning of an inference engine in parallel with a perception engine. On the one hand, an inference engine must be capable of performing the inferences provided in (Eq-4.54) along with the supporting concept algebra tools. On the other hand, a perception engine requires structures analogous to the layered structure employed by the Home UbiHealth model along with the real-time process algebra (RTPA) framework which is available in the designed model. Formally,

$$
\begin{aligned}
&CognitiveComputing = \\
&\quad CC\mathbf{PC} = \ (InferenceEngine\mathbf{PC}) \parallel (PerceptionEngine\mathbf{PC}) = \{ \\
&\qquad [(TestingInference\mathbf{PC}) \parallel (BehaviorProcessingRTPA\mathbf{PC})] \\
&\qquad \parallel [(Infrastructure\mathbf{PC}) \parallel (EventProcessingRTPA\mathbf{PC})]\}
\end{aligned}
\qquad \text{(Eq-4.55)}
$$

Autonomy is the property possessed by the model allowing it to present intelligent behavior provided that there are goals to be obtained based on decisions to be made. The decisions are based on evidences to make inferences that affect the achievement of defined goals. The model contains the Self-Controlling component, which includes a first-order control system that adjusts and regulates the status of the administered concepts of the model. The embedded control system function continuously tries to adjust the status of the involved concepts to achieve the set goals controlling the system formed by the concepts of the virtual level. Also, the Self-Controlling component includes a decision making element performing inferences on the output of the embedded control system. The inferences conclusions feed the physical environment whose reactions are supplied to the layered structure to act as the feedback mechanism to the conceptual system.

The behavior is the model according to [4.13] is classified into three classes:

1. the basic functional behavior, B_{basic}, depended upon the realized events, B_{event}, the time, B_{time}, based on periodic/aperiodic occurring events, and the occurring interrupts, $B_{interrupt}$.
2. the autonomous behavior, $B_{Autonomous}$, that depends upon the operation of the model to achieve specific goals, B_{goal}, and to perform inferences, , $B_{inference}$.
3. the presentation of cognitive behavior, $B_{Cognitive}$, presenting the functionality of making perceptions, $B_{perception}$,and inference, $B_{inference}$. .

The relation among the determined classes of behaviors is defined in [4.13] by $B_{time} \sqsubseteq B_{autonomou}, \sqsubseteq B_{cognitive}$.

Synopsis

The availability of the Denotational Mathematics framework provides the capability to describe formally the structure of a model's infrastructure revealing its static behavior. The formal description of the attributes and properties of the infrastructural components provides the capability to follow the dynamic behavior of the infrastructure. The operation of the infrastructure requires the support from medical ontologies in order to process and evaluate the developed information and the constraints from the patient's personal medical characteristics. The infrastructure's processing involves the development of a virtual context that corresponds to situations at home at the given moment upon which is applied control and made decisions. The infrastructure requires two interfaces, first, an interface for the full duplex communication with the medical devices with which the user interacts, and second, a bidirectional interface for the interaction with the medical professionals.

The system operates autonomously, following three steps. First, it senses raw data which converts into information upon which develops concepts. Second, the system develops in memory the virtual counterpart of the holding situation at home upon which applies control and makes decisions. Third, the system converts concepts into low level signals to control and guide the medical devices supporting the end-user. Also, the system uses Denotational Mathematics for knowledge representation as a function of the model's structure, the attributes and the properties of the structure's components, as well as, their interactions. In other words, the representation of medical knowledge is a function of the supporting system's structure and its capabilities.

The operation of the dispersed devices in the patient’s home environment needs the support from a system that is capable of providing the necessary intelligence in order to achieve semantic interoperability among the collaborating devices. However, the medical devices participate in ad-hoc, loosely coupled networks with frequent interleaves, incompatible devices, bounded memory and processing capabilities on each device, limited energy capacity, unexpected occurring events, and many sources of interference and noise. The supporting system must be capable of administering efficiently and effectively the raised issues achieving the aims concerning the patient’s health status.

CHAPTER 5. APPLICATION ISSUES

The operation of the designed model for the support of home ubihealth applications presumes the satisfaction of prerequisite conditions. The model must be capable of supporting the conditions related to user's mobility, which requires facilitating the applying principles of migration, polymorphism, and interoperability.

5.1 Migration

Migration is the procedures where the service from a running application cannot continue supporting the mobile user and control must be transferred to another compatible or incompatible device. Migration of the service must take place to uninterruptedly continue supporting the moving individual transferring control, data and code, to another device.

The model includes stationary and mobile devices, $d_{s_n}$and d_{m_k}, with n and k elements respectively. On the devices operate the application software a_i with i coexisting applications operating on the $d_r = (d_{s_n} \text{U}\ d_{m_k})$ devices. The a_i

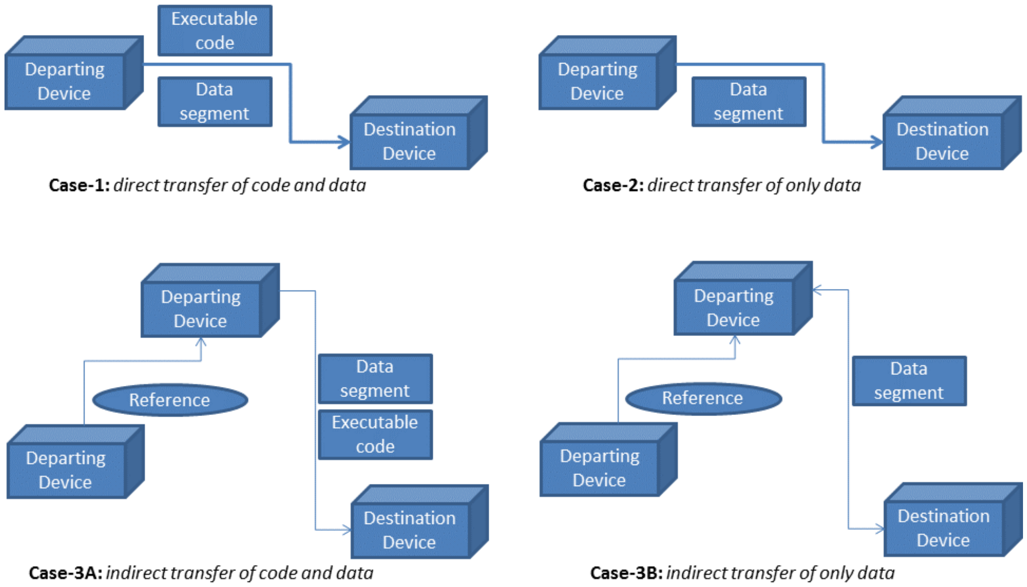


Figure-5.1 Available options for the applications' migration.

application performs discrete tasks t_p with $p \geq i$ allowing each application to perform many tasks. The performed tasks t_p on the d_r devices provide services $s_q = t_p$ x d_r consisting of the application software tasks and the devices functionality.

The migration of an application from a device d_{r1} to another device d_{r2} is performed by a task t_p seamlessly and without the intervention of the supported user. The departing device d_{r1}, by itself or assisted, must discover the most proper available device d_{r2} to be transferred with time to be constraining setting three options, as can be seen in Figure-5.1.

5.1.1 Typical Scenarios

The user is moving from the living room, ENV-A, to bedroom, ENV-B, see Figure-5.2. The supporting application realizes the user's move and sets available its data and code to the coordinating application. The coordinating application discovers the most proper device that is going to support the user towards ENV-B and receives data and code from the departing device and transfers the data and / or code to the arriving device.

The required infrastructure provides directory services to the operating devices, as shown in Figure-5.3, and the supporting application,$Device - \alpha,$ takes the

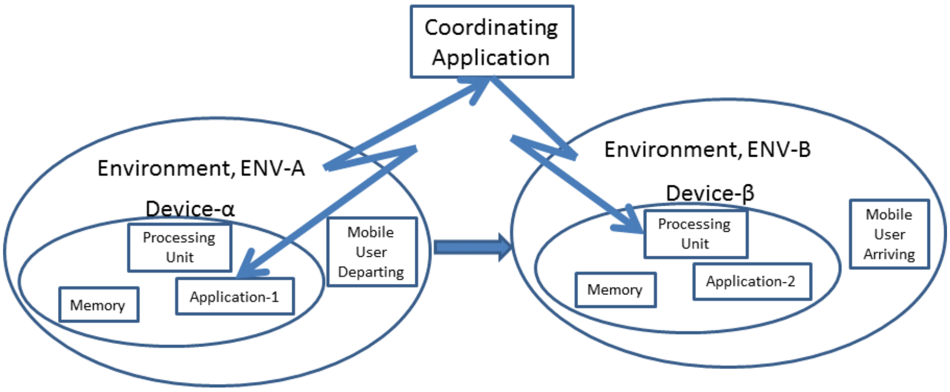


Figure-5.2 Application migration supported by a coordinating application.

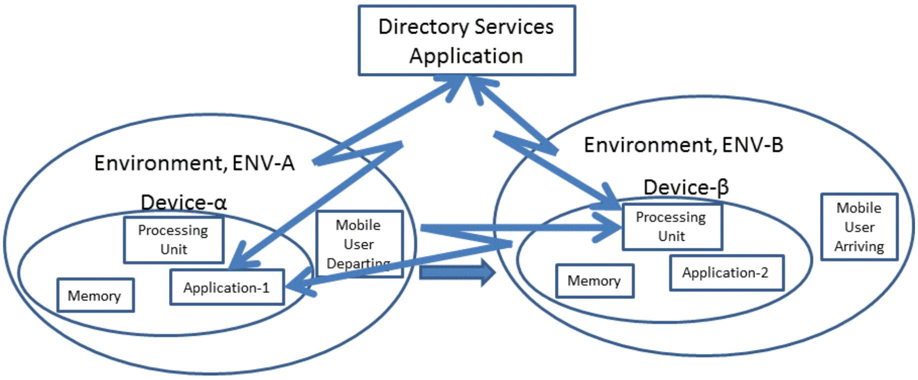


Figure-5.3 Application migration supported by negotiating devices.

responsibility to find the migrating device, $Device - \beta$, and initiate the migration process.

An event $e_{v1}(t)$ causes the replacement of the service $s_{k1}(t_{p1}, d_{q1}, t)$ by $s_{k2}(t_{p2}, d_{q2}, t)$ to continue supporting the user. Discovery and synchronization with the occurring event is tackled by the model, as shown in Figure-5.4. Discovery involves the destinations' availability, while optimization refers to the selection of the most adequate device.

5.1.2 Available models

Some of the available models suggested by the related research literature are listed below:

1. Aura Project: [5.1] is consisted of four parts, Figure-5.5, borrowed from [5.1].
2. Amigo Project: European (http://cordis.europa.eu/ist/) research project that adopts the SOA framework describing services independently from implementation and using web-services as components. Web-services are described by XML and

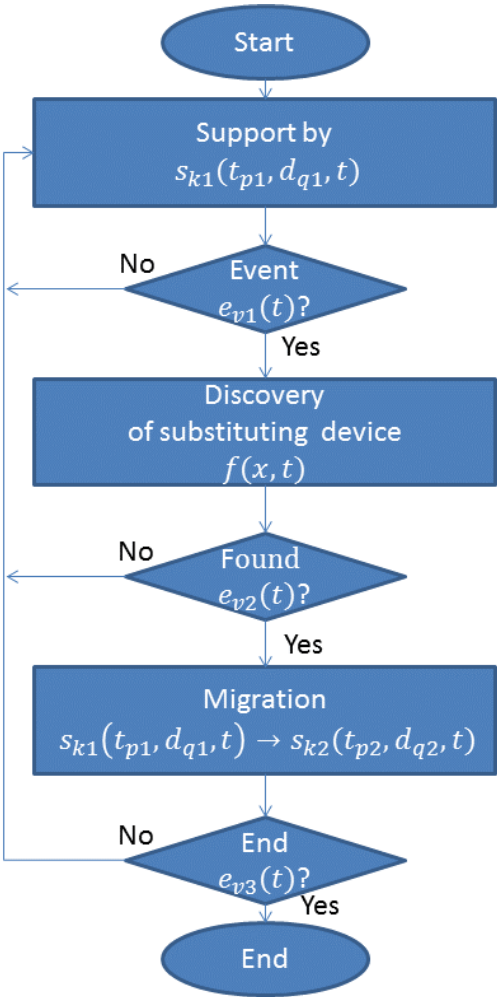


Figure-5.4 The rough flow of operations of a migration service.

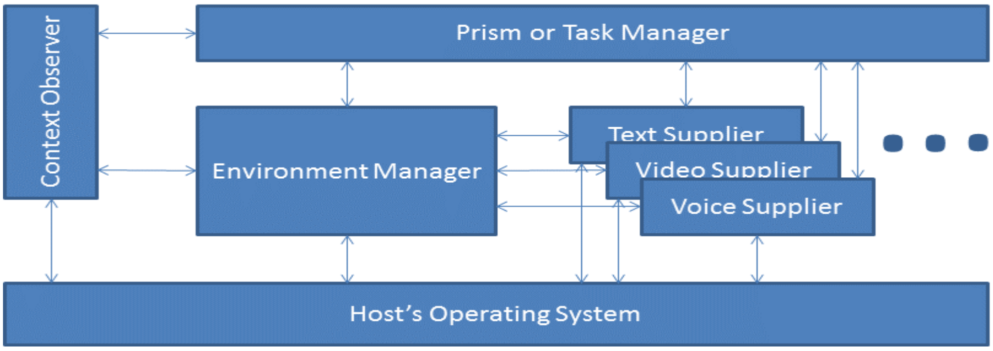


Figure-5.5 The Aura Project architectural building blocks.

expressed by WSDL communicating with SOAP through standard internet protocols, Figure-5.6, such as HTTL and SMTP achieving services' interoperability at application level [5.3, 4, 5].

3. Gaia Project: an Operating System using the concept of Active Spaces and the Model-View-Control paradigm [5.7], Figure-5.8, for each device, Figure-5.7.

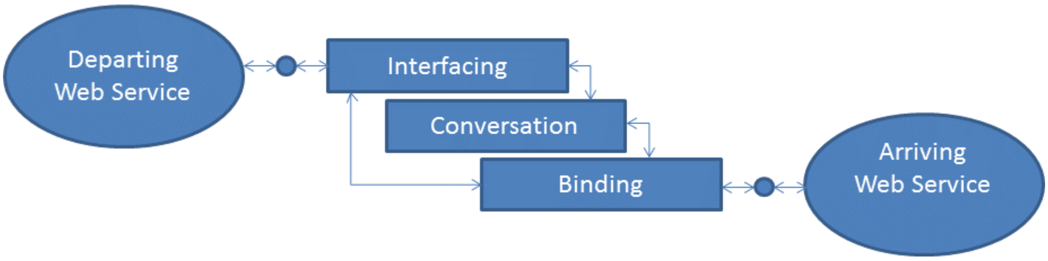


Figure-5.6 The Amigo consideration of web-services transfer software mechanism.

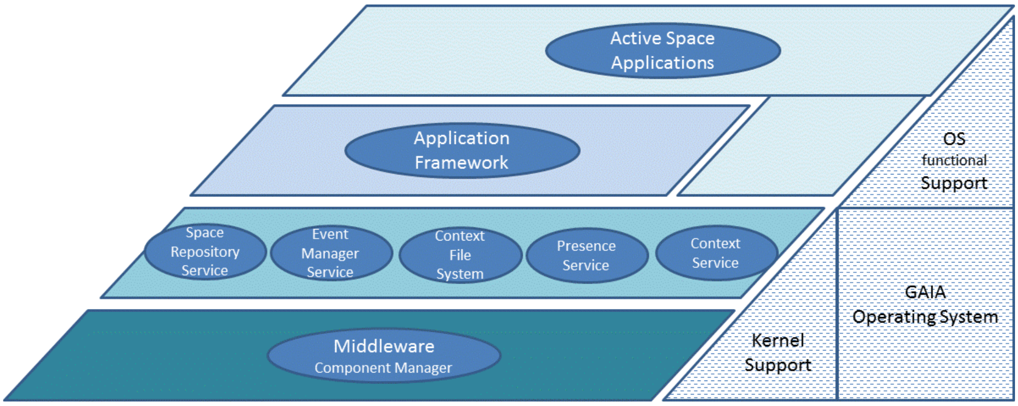


Figure-5.7 The components of GAIA Operating System's internal structure.

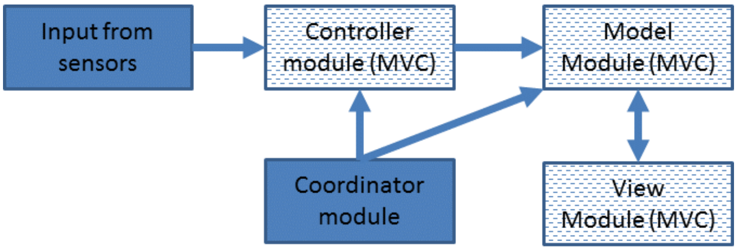


Figure-5.8 The Applications Framework constituting elements.

The model for context relies on quaternary atomic predicates in the form [5.6,7]: $Context(< ContextType >, < Subject >, < Relater >, < Object >)$.

4. Application Approaches with the OSGi Platform: the Open Services Gateway initiative (OSGi, www.osgi.org) describes a services platform accessible with Java providing registration and administration for the life-cycle of bundles, as illustrated in Figure-5.9.

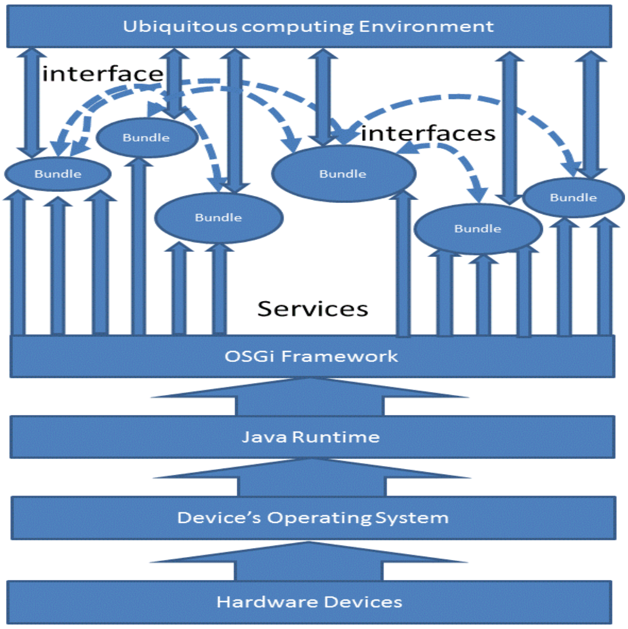


Figure-5.9 The OSGi framework exchanging services with the environment.

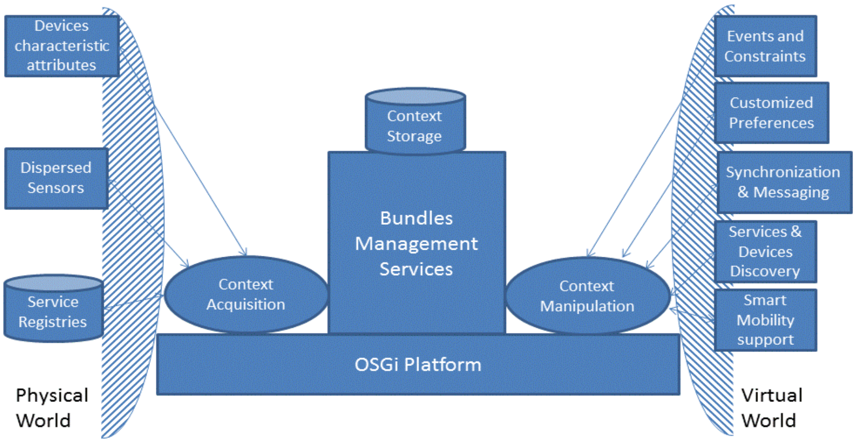


Figure-5.10 Building blocks of the COGITO [Char-5.8] enabling service.

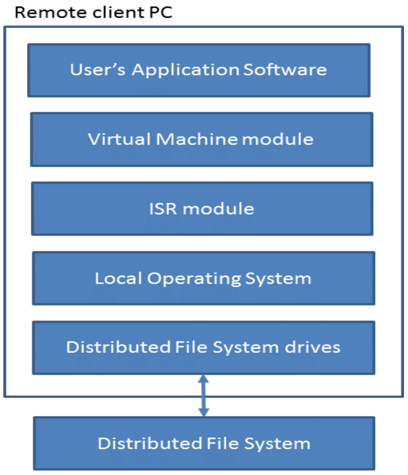


Figure-5.11 The ISR stack of software at the remote client personal computer.

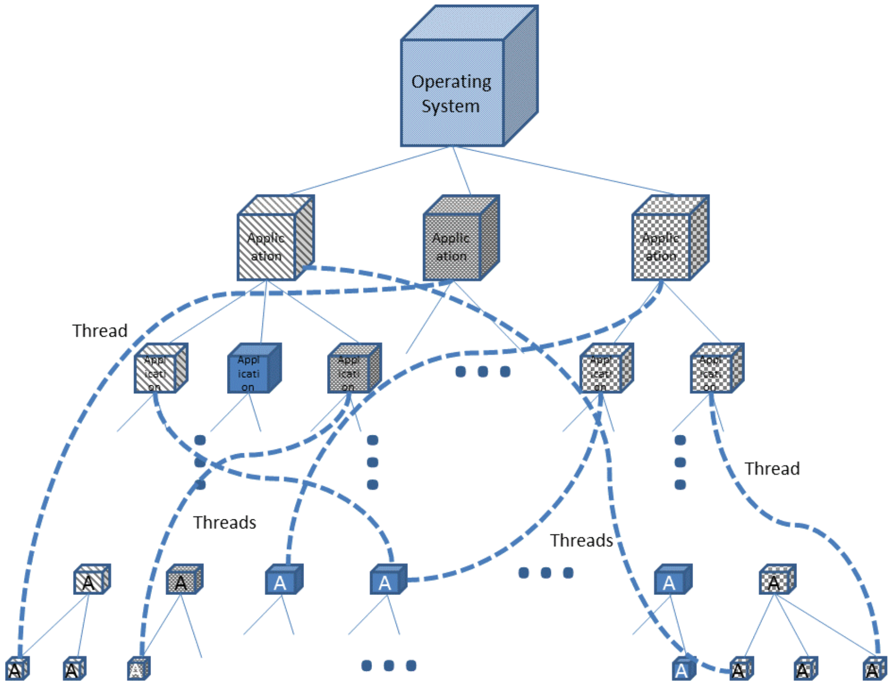


Figure-5.12 A threaded tree schematic to represent the involved one.world concepts.

The approach in [5.8] dispatches a service's state to all devices within the user's context. Figure-5.10 presents COGITO [5.8,9] enabling a structure providing context acquisition, storage, merging, and mapping of physical-virtual world.

5. ISR (Internet Suspend Resume mechanism) mechanism: employs the virtual machine and the distributed file system [5.10,11] employing internet to support mobility by suspending operation of applications at a location and resuming at another, see Figure-5.11.
6. One.world: discovers services which migrate to support the user's mobility based on [5.12,13,14,15]: context, ad-hoc services, and information sharing. In Figure-5.12 each node copies - calls applications, and relates with external events.
7. The TaskShadow approach: a framework mapping low-level services and high-level user's tasks using ontologies. The administration is performed using Petri-nets [5.14], Figure-5.13, for migration, tasks', service, and context management modules [5.15].
8. The HI3 approach: it is based on a layered architecture separating abstractions and allocating responsibilities to discrete modules [5.16], Figure-5.14.

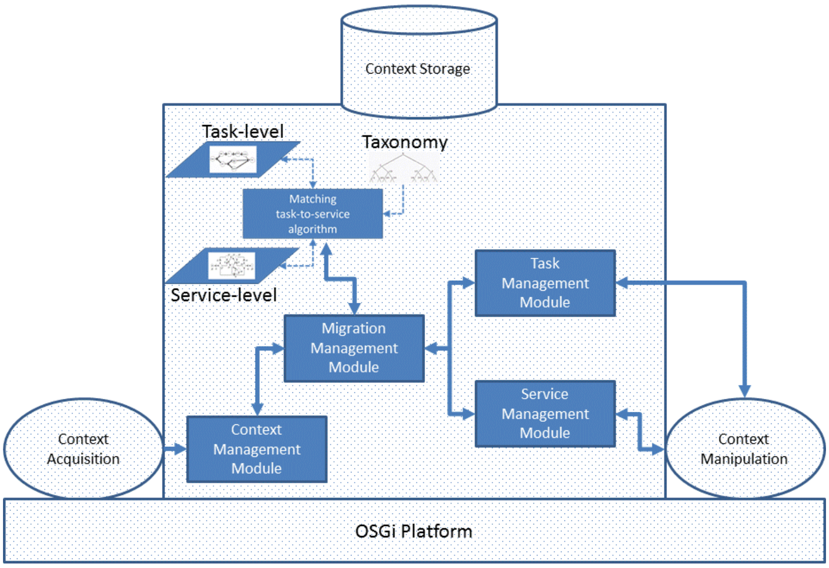


Figure-5.13 The TaskShadow architecture developed on top of the OSGi platform.

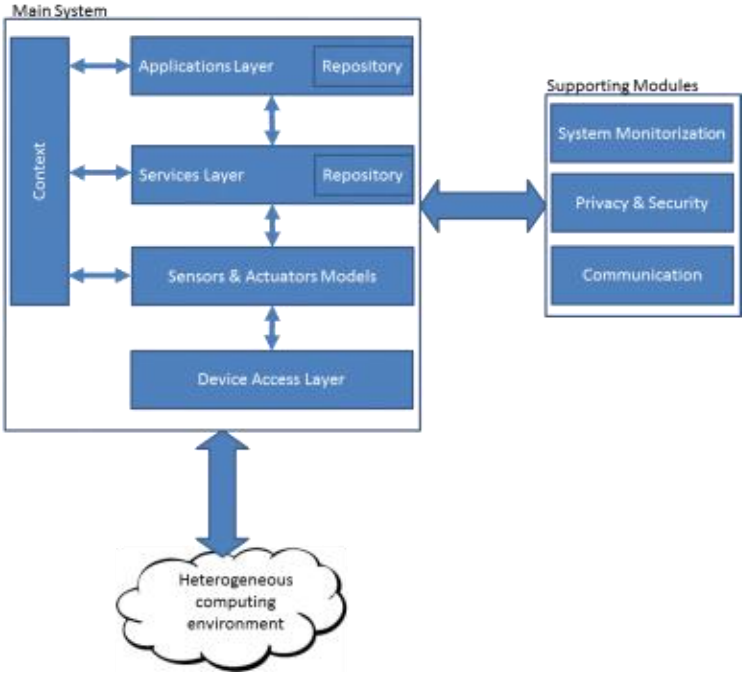


Figure-5.14 A block-diagram representation of the HI^3 architecture.

5.1.3 Migration Requirements

The requirements are specified by the medical and the operational characteristics:

A. Functionality

a. Functional requirements: refer to the behavior of the model and include:

  i. Figure-5.15, a hierarchical structure referencing activities with varying abstraction.
  ii. Figure-5.16 provides a vector representation of the interrelationships of physical magnitudes of space, time, and domains and the levels of activity abstraction.
  iii. identification of persons and a naming system to give identities to all participating persons and entities, tangible and intangible, Figure-5.17.

b. Non-functional are distinguished into two parts: the participating stakeholders and the required codifications. Specifically,
  i. Stakeholders: refers to those of the authorities and those professional roles that make the Home UbiHealth model effective and feasible including the Healthcare System, the Government, the Market, the Insurances, the Health Funds.
  ii. Codifications: refers to complete sets of structured data issued and maintained by specializing authorities. Such sets provide detailed identification and precise description of necessary data for the operation of the Home UbiHealth model.

c. Domain requirements comprise the characteristics of the underlying medical requirements for the support of the individual at home falling into three categories: Personal care and well-being, Primary care, and Secondary care medical services.

The software applications must present the following characteristics: Security and privacy, Determinism, Reliability, Continuity of flow, Adaptation, Fault tolerance, Scalability, Synchronization, Modularity, Connectivity, Reusability, and Proactivity.

B. Prerequisites: the functionality of the software applications poses the property of service which includes the following:
  a. Roles of Producer, Registries, and Consumer.
  b. Services with characteristics of quality, predictability, and promptness.
  c. Objectives referring to goals, accountability, and relationships.
  d. Operations discovery, composability, adaptation, and continuity.

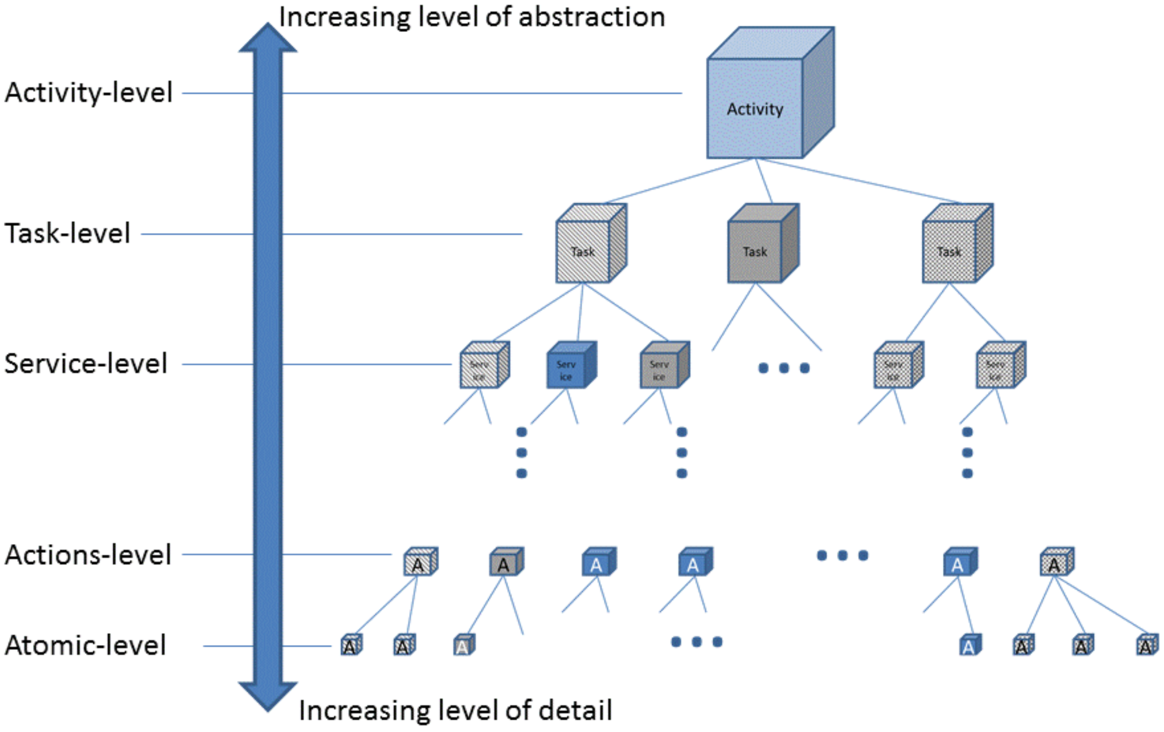


Figure-5.15 A hierarchical structure presenting variable level of activities abstraction.

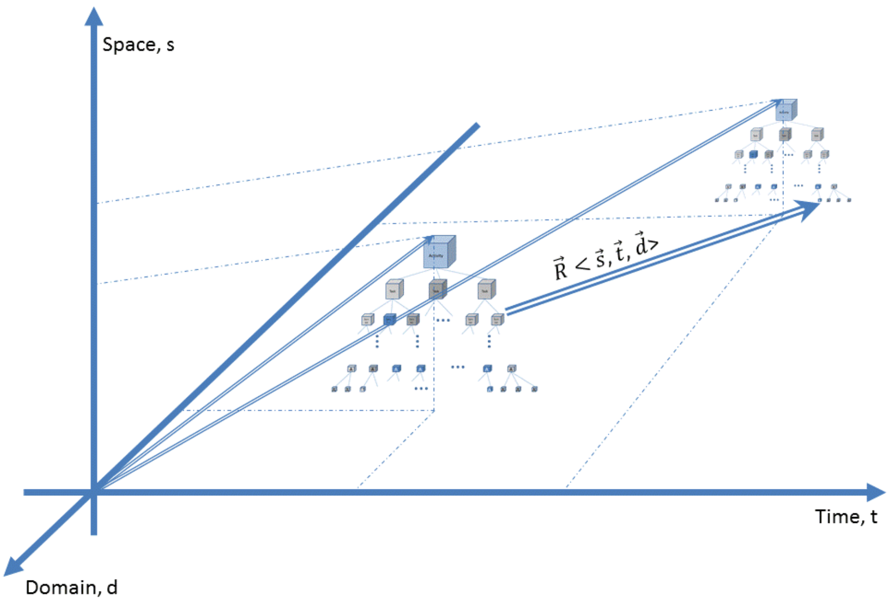


Figure-5.16 A vector representation of the involved physical magnitudes.

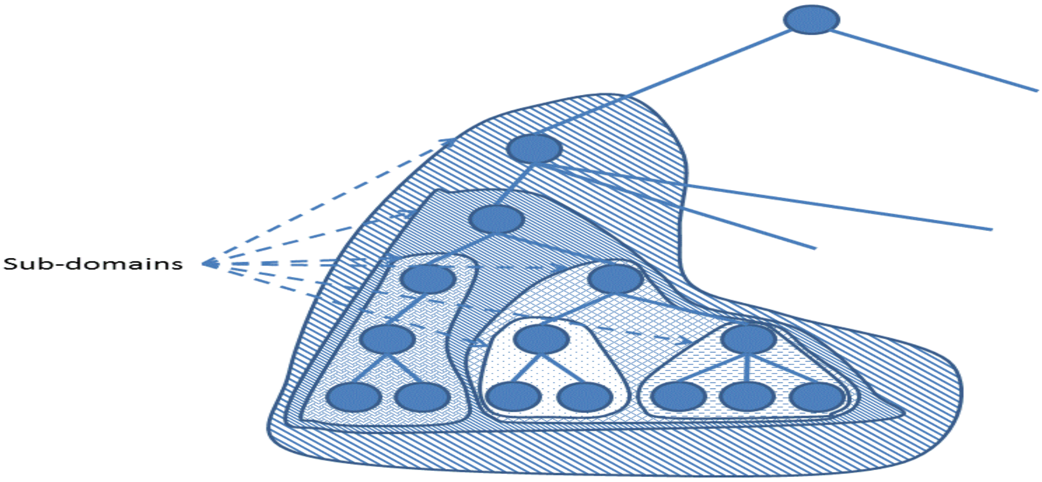


Figure-5.17 A naming system and sub-domains representation.

5.2 Polymorphism

The property of software applications to reform their internal structure preserving the same functionality is considered as polymorphism. As the user changes locations, different sets of devices provide support for the determined tasks. In other words, the set of devices monitoring and preserving the health status at the living room can be different than the set of devices supporting the individual at the kitchen. Hence, control transfer or migration must take place transferring at least the state, i.e., the data set, of the supporting application from one location to another at a different device. The user's mobility causes migration which, in turn, causes the use of polymorphism because the state of the supporting application must be transferred at another device, at another location within the house. If the departing and the destination devices are of the same type, then, only the state of the application must be transferred to the destination device which means that the values of the application's parameters must be transferred. In cases where the departing and the destination devices are inhomogeneous then polymorphism must be applied transferring a properly modified program along with its data.

5.2.1 Prerequisite for Autonomy

Migration must take place spontaneously and without the user's intervention and it must be decided by the supervising system involving the following set of operations:

1. Activation: it is based on evidences from a variable or a set of parameters.

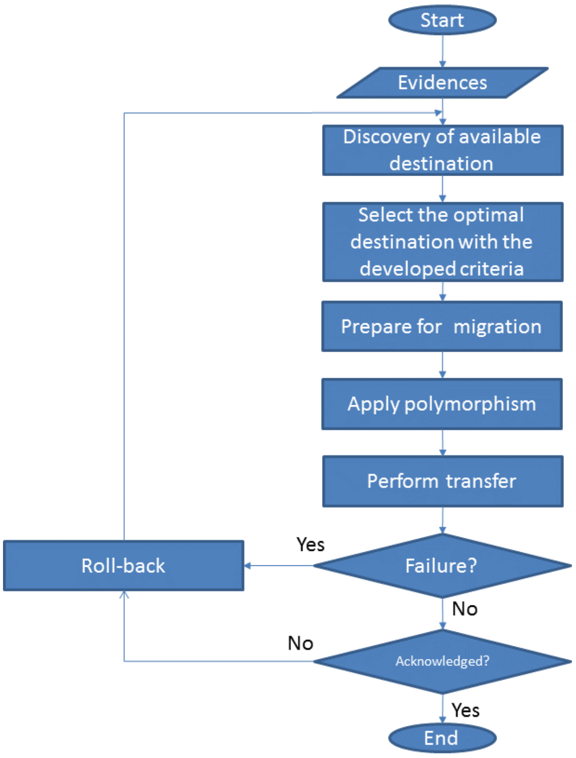


Figure-5.18 The procedure for the migration of software applications.

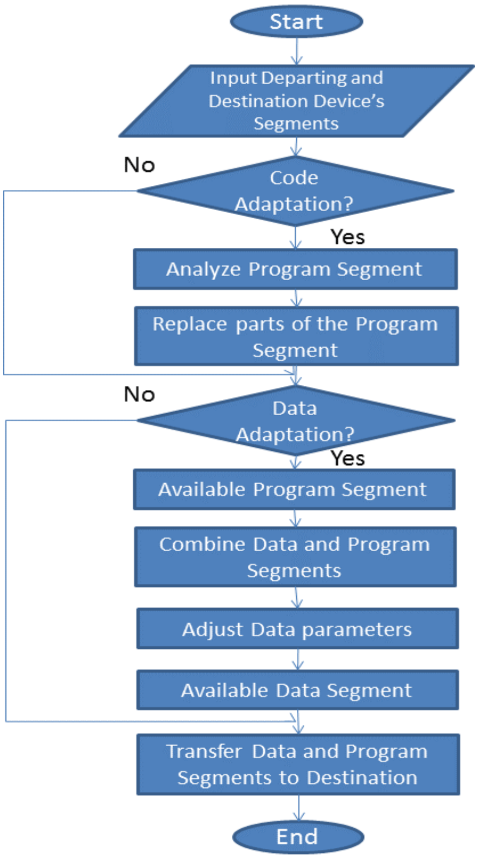

Start
Input Departing and Destination Device's Segments
No
Code Adaptation?
Yes
Analyze Program Segment
Replace parts of the Program Segment
No
Data Adaptation?
Yes
Available Program Segment
Combine Data and Program Segments
Adjust Data parameters
Available Data Segment
Transfer Data and Program Segments to Destination
End

Figure-5.19 The flow of operations for the migration process.

2. Discovery: it refers to the available destinations with two options:
    a. Self-discovery: it is aware about the availability and the IP address of the destination.
    b. Assisted discovery: it receives external assistance, e.g., from a dedicated agent.
3. Selection: it considers the applied policies performing optimization with respect to defined parameters, the similarity of the destination's computing environment, the time required for migration considering the transferring volumes, the available bandwidth, and the expected user's locations.
4. Preparation: it is about the process of storing the current state at the output buffers of the device.
5. Polymorphism: it transfers the application using the results of optimization concerning:
    a. Data migration: it transfers the data segment of the departing device.
    b. Data and program migration: there are two options:
        i. The application modifies its content to meet the requirements at destination.
        ii. The application replaces the program and data segments with compatible others.
6. Transfer: there are two alternatives depending on the location and the destination device.
    a. Direct transmission of the program and the data segments to the destination.
    b. External assistance is provided to control the transmission.
7. Acknowledgement: it notifies by a signal the operation's proper completion.
8. Roll-Back: it resumes the transfer or makes different decisions allowing reconsiderations.

The migration procedure is depicted in Figure-5.18 where polymorphism is applied autonomously based on the optimization of the available criteria.

5.2.2 Applied Polymorphism

The limited availability of computing resources of an ordinary device in the Home UbiHealth model requires the computer processing to be performed by a supporting infrastructure. There exist the following two migration cases for an application’s transfer also illustrated in Figure-5.19:

1. Transfer of Data Segment: when the departing and the destination devices are compatible.
2. Transfer of Data and Program Segments: when the departing and the destination devices are heterogeneous and the program segment must be properly structured to meet the requirements of the destination device and data segment adapted accordingly.

Depending upon the adopted software technology, the replacement of parts of the migrating software code can be performed in the following ways [5.18]:

1. Parametric: it adjusts the departing device’s data segment.
2. Inclusion/overriding: it is the case where the destination device’s computing environment requires the replacement of parts or the entire program segment by a set of program routines without recompilation of the program’s code.
3. Overloading: it is about the changes in the internal structure and contents of the program’s segment to meet the computing requirements of the destination’s environment. A different mix of subroutines is composed preserving the same functionality.
4. Coercion: it refers to the administrative infrastructure when it replaces the entire program segment of the departing device with an equivalent program segment adapting the data segment in order to achieve the same functionality at an identical semantic state.

5.2.3 Requirements for polymorphism

The Home UbiHealth model must be capable of supporting without interventions the property of mobility. The operation of the model requires a supporting framework that is aware of the information regarding the state of all devices operating in the home environment, aware about the location and the relations among all entities, it can make decisions based on the acquired information to provide continuous support, and it can support migrations to achieve its goals. Hence, the system must present the characteristics of autonomy and polymorphism within an appropriate framework.

5.3 Adjusting Control

The layered framework of the Home UbiHealth model is supported by positive and negative feedback mechanisms to adjust the controlling parameters.

5.3.1 Negative Feedback mechanism

It refers to Mechnism-10 of Figure-3.11 which provides the virtual entities' values, relations, and functionality back to the physical level. The administration of smart devices requires tools common in Control Theory and cognitive approaches to manage medical aspects. From a Control Theory point of view, the available testing tools include the Observability, the examination of the output in order to provide the expected input values, and Controllability tests, as well as the examination of the input values of a system to provide the expected output.

Mechanism-10 must test in advance the future states of the system since some of them are acceptable while others must be avoided. Mechanism-10 can bring the system at certain desirable states based on the evidences provided by the tools such as a Petri-net which can accomplish by the visualization. Hence, the tool of Reachability is needed to test the capability to reach desired states. An arbitrary use of an eigenvalue that satisfies the Controllability and Observability tests can be further used to define the Stabilizability, which it ensures that the system is in desirable states, and the Detectability, which it asserts that the system states are observable and eventually reachable. Hence, there is a complete set of tools for the system's administration.

5.3.2 Positive Feedback Mechanism

The positive feedback mechanism sets some suggested values and at a later stage finds out if succeeded to achieve the desired state. In Figure-3.11, Mechanism-9 plays a dual role, as a (i) positive feedback controller, and (ii) a processor feeding with data the rules-based system. Mechanism-9 must have an interface to receive raw data from the physical layer. Mechanism-9 accesses a rule-based system, updating and redefining rules with the forward or the backward chaining method. Forward chaining or data-driven reasoning refers to the operations that form rules given a set of values representing conditions and the values representing the counterpart of the conclusions. Backward chaining or goal-driven reasoning refers to the operations performed developing rules to use a set of given goals. The tree of rules involves (i) evidence nodes, providing data operators, (ii) external nodes, providing calls to procedures, and (iii) reference nodes, forming meta-rules to test policies.

5.3.3 Mechanisms Requirements

The required feature of adaptation must be performed in an autonomous manner which means that the system must make its own decisions. A formal description of the supporting system must be provided describing formally the features of autonomy and the features of adaptation. In this direction, the design principles of separating the concerns must be supported in order to have the capability of developing software applications facing one problem at time. The development of software applications must support the encapsulation of compound attributes and properties without referring to each individual device.

Synopsis

The collaborating inhomogeneous and incompatible medical devices in the Home UbiHealth environment are guided by a supporting computing infrastructure. The computing infrastructure provides adequate support to the operating medical devices administering the transfer of data and programs' code between devices achieving continuity of the user's medical support. The programs' migrations between medical devices require the proper administration of the carried procedures between a sender device and a receiver device. In some cases, the structure of the migrating programs must be properly adjusted to meet the requirements by applying the principle of polymorphism. Such a principle allows the formed system to operate autonomously provided the availability of the necessary decisions-making mechanisms.

The synchronization and the governance of the medical devices operational behavior is obtained by the application of control principles and methods. Control is focused on the patient with the adoption of properly designed controllers. The patient is considered to be part of the controlling plant of the formed control system while the role of the observer is shared among the treating medical professionals and the patient demanding the satisfaction of personal preferences. In other words, the formed system features both internal and external observers too. Therefore, the upholding infrastructure takes over the responsibility for the guidance of the dispersed medical devices to meet the medical requirements and the personal preferences of the supporting patient in the Home UbiHealth environment.

CHAPTER 6. CONCLUSIONS

The Home UbiHealth model is described using Denotational Mathematics providing the necessary independence to administer with algebraic means the conceptual entities. Furthermore, the algebraic manipulation of concepts provides the capability to administer complicated medical systems in unstructured environments such as that of home. In the following paragraphs, the main achievements of this thesis are presented along with criticisms about the benefits attained, a brief discussion about the designed model, and suggestions for future research.

6.1 Results

The formal description of the Home UbiHealth system with Denotational Mathematics provides the capability to design formal models for the provision of healthcare services at home in a ubiquitous computing environment. The impact of the formal design of the model has the following impacts:

(a) Results: the design considers the model of an open system including the entire healthcare environment including the informal part of Primary Healthcare which is home healthcare. The surrounding healthcare environment is a large framework consisted of roles and authorities at the community, the regional, and the national levels. The roles are comprised of the medical professionals, the nursing personnel, the communities' volunteers, and the family members. Among the participating authorities are healthcare and governmental organizations such as healthcare centers, hospitals, social security, health funds, insurance, and the members of the market. The participating roles and authorities in the model form a complicated system with the aim to support the individual at home. The supported individual is part of the model forming a self-referenced system considering the control of the status of the individual’s health. The complicated system is described formally with Denotational Mathematics providing the adequate means to monitor, control, and regulate the model. The design achieves the autonomous functioning of the model using conceptual constructs instead of the analytical and detailed values of signals from the dispersed devices in model’s environment. The involvement of Denotational Mathematics provides the means to manipulate algebraically defined artificial constructs such as processes, concepts, and knowledge in order to personalize the treatments and the provision of medical services at home.

(b) Usage: the model presents an internal infrastructure to accommodate, facilitate, and manipulate the involved processes. Taking advantage of the Denotational Mathematics component of Process Algebra provides the capability to apply algebraic operations on the used constructs of processes, concepts, and knowledge. The model’s infrastructure involves a layered structure supported by peripheral components. The layered structure is employed for the transformation of raw data into context, then turn context into information, use information to develop context-awareness, and then, transform context-awareness into processes, processes into concepts, and concepts into knowledge. The concepts and the represented knowledge reside onto the virtual level of the layered structure. The four employed peripheral components operate in such a way to provide, first, the interface with the devices and the entities of the surrounding (Event Driver), second, the component through which the medical and personal constraints are introduced to the model (Constraint Component), third, the required taxonomies representing the required knowledge (Knowledge Component), and forth, the controlling and regulating component equipped with controlling tools and decision support facilities (Self-Controlling Component).

(c) Model’s Capabilities: the structured layer component of the model transforms raw signals into concepts. The model employs Denotational Mathematics tools to manipulate conceptual constructs using Real-Time Process Algebra (RTPA), Concept Algebra, System Algebra, and Inference. Thus, the layered structure of the model transforms raw data into concepts with which it is performed the administration of the model and, consequently, the status of the individual’s health. Hence, instead of using a large number of detailed data from the dispersed devices or the sources of data from outside of home, the model uses the corresponding set of interrelated concepts with the analogous values using the available algebras. The model applies control on the set of the developing concepts which correspond to the received signals of the occurring context. Depending upon the formed relations among concepts, the model outputs data values directed to the connected dispersed devices in order to allow the system to achieve its goals based on the decisions made satisfying medical criteria. Hence, the model uses concepts in order to control the connected ubiquitous computing devices in order to achieve its goals with the decisions made acting autonomously.

(d) Benefits: from the medical point of view, the model offers the opportunity to completely personalize the applied treatments and the followed cures. The autonomous behavior of the model supports the individual’s health status with a systemic approach based on the availability of medically defined goals supported by control tools and the capability to perform knowledgeable decisions to achieve the set goals. The individual’s health status is characterized by a set of parameters providing the capability to consider the state of the individual’s health as a system manipulating complex medical relations. The model attempts to introduce formally the minimal standardization to the informal part of Primary Healthcare supporting the provision of healthcare at home. Also, the processes carried out by the medical professionals are different than the way it was provided in the past taking advantage of the ubiquitous computing characteristics for the patient’s treatment at home. Moreover, the model provides a premature cognitive computing environment with inference along with the perception capabilities of systems with autonomous capabilities.

6.2 Evaluation

The purpose of the Home UbiHealth model is the medical control of an individual’s health status. The complexity of the model increases considering all the co-existing parameters related with cultural issues, social conditions, economic prerequisites, political impacts, technical requirements, and specific necessities of medical treatments. The evaluation of the design of the Home UbiHealth model must be based on defined criteria and applied methods while the achievement of the set goals can either be described with quantitative and qualitative characteristics. The performance of the designed model depends, on the one hand, on the number and the type of sensing devices, and on the other hand, on the kind of actuating devices. In other words, the system can respond to those concepts that can be sensed and can be affected by the operation of the model. The model requires the employment of proper interfacing modules in order to achieve the capability to communicate with the devices in the ubiquitous computing home environment. The raised issues and challenges related by the introduction of an operational Home UbiHealth model can be classified as follows:

- Socio-political: it changes the social relations among the healthcare stakeholder and the followed processes carrying out restructured healthcare procedure with wider impacts on human and consumers behaviors with respect to Social Security, Insurance, Healthcare Industry, and practicing Medicine.
- Practicing Medicine: it provides plenty of accurate and, most probably, real-time evidences about the individual’s medical treatment at home.
- Medical Professionals Collaboration: the healthcare services are provided by a collaborating and synchronized team of professionals with discrete roles in the medical care of the individuals’ at home revealing the need for the introduction of new professional roles for caregivers.
- Learning and training: the model facilitates the means to provide formal medical treatment at the informal part of the Primary Care of any Healthcare System which is the medical cure at home. The medical professionals must include in their training the use of the advantageous tools provided by the model in their continuous battle against disease and death.
- Outcomes predictability: the designed model refers to a stochastic system in spite of the fact that its constituting infrastructure is developed with deterministic components. The stochastic characteristics are provided by the involvement of the supported individual, with declared personalized preferences, as a component of the formed system, as well as, the consideration of the medical constraints placed by the charged medical doctors and the associated medical knowledge. The use of protocols,

treatments, and therapies tend to reduce the impact and the scope of the stochastic influences but still dominates in most of the cases.

- Real versus Virtual: the occurring processes at the physical environment are sensed, examined, analyzed, and tested within the conceptual infrastructure of the model. The model aims to develop its augmented virtual counterpart upon which all decisions are made in order to succeed in its mission supporting the individual’s health status at home. The virtually developed conceptual model of the occurring reality contains many relations and properties that it is impossible to be revealed by the functioning sensors at the physical environment. Thus, some sensed events can be completely independent at the physical level while they can be associated and influenced at the virtual level as a function of the holding level of scientific knowledge.
- Medical Knowledge: it dominates the operation of the designed model affecting directly the type and kind of set goals to be achieved by the system, the sort of decisions made in order to achieve the set goals, and the variety of controlling parameters in the applied therapies. Using the representation of medical knowledge, the signals and raw data received from the dispersed devices in the ubiquitous computing home environment are transformed into complete information, consistent knowledge, and usable concepts upon which can be made intellectual decisions.

The operation of the Home UbiHealth model is based on the reception of signals and raw data that, potentially, face problems related to the functioning environment as well as the nature and the quality of the used devices. The surrounding operational environment can introduce white noise at the operation of the model affecting the fidelity of its efficiency. Similarly, the mobile devices and the limitation of the supplied energy can cause abrupt failures that set the designed model incapable to operate properly. Also, the applied noise on the operation of the dispersed devices, e.g., the nearby operation of a motor from the home air-conditioning system, can cause malfunctions to the devices that supply misleading or fluctuating data to the modeled system. Moreover, the level of difficulty is augmented by the consideration of the systematic errors included in the operation of the model introduced by the influences received by the human interventions in the operation of the system such as the placement of medical constraints or the declaration of personal preferences from the supported individual.

In spite of the difficulties and the hostility of the home environment, the model introduces the adequate level of standardization required to support the individual’s health status. The designed model can cope with ambiguity and imprecision introduced by the sensing network of devices by filtering them out using fuzzy and fusion techniques with the Dempster-Shafer Theory of Evidences due to the critical roles of time. Missing readings from the physical environment can cause serious problems to the operation of the model that can either increase the sampling rate to receive vital data the soonest possible or make inferences based on the available set of data.

The appropriateness of the model functionality is based on the availability of encoded medical knowledge. The designed model allows the medical professionals to place constraints expressing the ordered therapies within the knowledge framework formed by the associated ontologies. The model bases its decisions upon the soundness and completeness of the available medical ontologies and their specialized taxonomies. The set goals and the performed decisions are both based on the collaborating ontologies forming a closed intellectual sub-system of predictable and limited responsiveness to the limits of the encoded medical knowledge. The embedded markov chains provide the required support for the required systemic behavior of predictability providing the necessary means to medical professionals and the supporting caregivers to base their therapeutic activities on evidences.

The Home UbiHealth model can use a large set of sensed signals and raw data which are transformed into the corresponding set of interacting concepts of reduced cardinality. The availability of the virtual counterpart of the real physical environment provides the capability to vary the virtual representation of the physical level depending upon the availability of the sensed data. Thus, testing the achievement of goals and making decisions can be based on a smaller set of data at a higher level inference. Hence, the model can perform on varying availability of sensing devices. Also, the output of the designed model can drive the actuating devices of the physical environment supporting autonomously the required migration of running software applications from a device to another backing the individual’s mobility. The functional behavior of the model bearing the administration of critical time operations, the individual’s mobility, and the designed system’s autonomy shows the adequate efficiency of the system to sustain the individual’s health status at home with the necessary evidences.

6.3 Future Work

The research achievements from the design of the Home UbiHealth model offer the following directions for further research:

(a) Medicine: standardization of the available protocols, therapies, and treatments supplying ontological entities with specialized scientific knowledge. Such an effort must provide the proper parameterization in order to be administered by the available level of computing processing. Also, the required parameterization is going to provide the capability to personalize treatments and medical therapies to the needs of the individual.

(b) Cognitive Computing: the exploration of the capability to perform intelligent inferences and acquisition of perception mimicking the execution of intellectual operation analogous to those of human brain. Also, the design of the Home UbiHealth model can be enriched with a concepts parser to examine and analyze the medical semantic status of the individual's health associated with the available medical ontologies.

(c) Home Healthcare: the formal incorporation of home healthcare into Primary Care of a Healthcare System. The designed model provides a formal description that promotes the collaboration of Healthcare and Social Security.

(d) Security and Privacy: taking advantage of the approaches of ubiquitous computing environment at home and Internet of Things (IoT) raises issues related to the protection of security and privacy at home. The designed model must be supported by methods that ensure the proper access to the private health information while permitting the remote collaboration of the needful medical professionals.

REFERENCES

Executive Summary

[1] Wang, Y. (2002). The real-time process algebra (RTPA). Annals of Software Engineering, 14(1-4), 235-274.

[2] Wang, Y. (2008). RTPA: A Denotational Mathematics for Manipulating Intelligent and Computational Behaviors, Int'l Journal of Cognitive Informatics and Natural Intelligence, 2(2), 44-62, April-June 2008

[3] Wang, Y. (2008). On concept algebra: A denotational mathematical structure for knowledge and software modeling. International Journal of Cognitive Informaticsand Natural Intelligence, 2(2), 1.

Chapter-1

[1.1] Weiser, Mark, Rich Gold, and John Seely Brown. "The origins of ubiquitous computing research at PARC in the late 1980s." *IBM systems journal* 38.4 (1999): 693-696.

[1.2] Weiser, Mark, and John Seely Brown. "The coming age of calm technology."*Beyond calculation*. Springer New York, 1997. 75-85.

[1.3] Weiser, Mark, and John Seely Brown. "Designing calm technology." *PowerGrid Journal* 1.1 (1996): 75-85.

[1.4] Weiser, Mark. "Connecting the physical world with pervasive networks." (2002).

[1.5] Van Laerhoven, Kristof, et al. "Medical healthcare monitoring with wearable and implantable sensors." *Proc. of the 3rd International Workshop on Ubiquitous Computing for Healthcare Applications*. 2004.

[1.6] Chan, Marie, et al. "Smart homes—current features and future perspectives."*Maturitas* 64.2 (2009): 90-97.

[1.7] Marschollek, Michael, et al. "Sustainable ubiquitous home health care-Architectural considerations and first practical experiences." *Studies in health technology and informatics* 129.1 (2007): 8.

[1.8] Shabo, A. "AGlobalSocio-economic-medico-legal Model for the Sustainability of Longitudinal Electronic Health Records." *Methods Inf Med*45 (2006): 240-5.

[1.9] Bott, O., et al. "Introduction: Ubiquitous ComputinginHealthCare." *Methods Inf Med* 44 (2005): 473-9.

[1.10] Mantovani, Eugenio, et al. "eHealth to mHealth–A Journey Precariously Dependent Upon Apps?." *European Journal of ePractice* 21 (2013): 48-66.

[1.11] Dawadi, Prafulla N., Diane J. Cook, and Maureen Schmitter-Edgecombe. "Automated cognitive health assessment using smart home monitoring of complex tasks." *Systems, Man, and Cybernetics: Systems, IEEE Transactions on* 43.6 (2013): 1302-1313.

[1.12] Nidumolu, Ram, Coimbatore K. Prahalad, and M. R. Rangaswami. "Why sustainability is now the key driver of innovation." *Harvard business review*87.9 (2009): 56-64.

[1.13] Marschollek, Michael, et al. "Sustainable ubiquitous home health care-Architectural considerations and first practical experiences." *Studies in health technology and informatics* 129.1 (2007): 8.

[1.14] Hilty, Lorenz M., Claudia Som, and Andreas Köhler. "Assessing the human, social, and environmental risks of pervasive computing." *Human and Ecological Risk Assessment* 10.5 (2004): 853-874.

[1.15] Friedewald, Michael, and Oliver Raabe. "Ubiquitous computing: An overview of technology impacts." *Telematics and Informatics* 28.2 (2011): 55-65.

[1.16] Mucheleka, Martin, and Raija Halonen. "ERP in Healthcare."

[1.17] Prajapati, Biraj, et al. "Review of e-Health Frameworks." (2015).

[1.18] Bouamrane, M. M., C. Tao, and I. N. Sarkar. "Managing interoperability and complexity in health systems." *Methods Inf Med* 54.1 (2015): 1-4.

[1.19] Gvozdanović, Darko. "Interoperability within eHealth arena."

[1.20] Wickramasinghe, Nilmini. "Pervasive computing and healthcare." *Pervasive Health Knowledge Management*. Springer New York, 2013. 7-13.

[1.21] Silva, Bruno MC, et al. "Mobile-health: A review of current state in 2015."*Journal of biomedical informatics* 56 (2015): 265-272.

[1.22] Gagnon, Marie-Pierre, et al. "m-Health Adoption by Healthcare Professionals: A Systematic Review." *Journal of the American Medical Informatics Association*(2015): ocv052.

[1.23] Treacy, Ceara, Fergal McCaffery, and Anita Finnegan. "Mobile Health & Medical Apps: Possible Impediments to Healthcare Adoption." (2015): 199-202.

[1.24] Krishnan, Sharanie Banu, and Jaspaljeet Singh Dhillon. "Barriers to Adoption of Consumer Health Informatics Applications for Health Self-Management.", 2015

[1.25] Harper, Richard, ed. *Inside the smart home*. Springer Science & Business Media, 2006.

[1.26] Augusto, Juan Carlos, et al. "Enhanced healthcare provision through assisted decision-making in a smart home environment." *2nd Workshop on Artificial Inteligence Techniques for Ambient Inteligence*. 2007.

[1.27] Mendes, Tiago DP, et al. "Smart home communication technologies and applications: Wireless protocol assessment for home area network resources."*Energies* 8.7 (2015): 7279-7311.

[1.28] Serna, Maria, Cormac J. Sreenan, and Szymon Fedor. "A visual programming framework for wireless sensor networks in smart home applications." *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on*. IEEE, 2015.

[1.29] Edwards, W. Keith, and Rebecca E. Grinter. "At home with ubiquitous computing: Seven challenges." Ubicomp 2001: Ubiquitous Computing. Springer Berlin Heidelberg, 2001

[1.30] Ye, Xiaojing, and Junwei Huang. "A framework for cloud-based smart home."*Computer Science and Network Technology (ICCSNT), 2011 International Conference on*. Vol. 2. IEEE, 2011.

[1.31] Gröne, Oliver, and Mila Garcia-Barbero. "Integrated care: a position paper of the WHO European Office for Integrated Health Care Services." *International journal of integrated care* 1 (2001).

[1.32] Canadian Institute for Health Information. National Consensus Conference on Population Health Indicators Final Report. Ottawa:CIHI, 1999.

[1.33] American Academy of Pediatrics Ad Hoc Task Force on Definition of the Medical Home. The medical home. Pediatrics. 1992;90:774.

[1.34] Davidson EJ, Silva TJ, Sofis LA, Ganz ML, Palfrey JS. The doctor’s dilemma: challenges for the primary care physician caring for the child with special health care needs. Ambul Pediatr. 2002;2:218–223.

[1.35] McManus, M., White, P., Barbour, A., Downing, B., Hawkins, K., Quion, N., ...&McAllister,J.W.(2015). Pediatric to adult transition: a quality improvement model for primary care. Journal of Adolescent Health, 56(1), 73-78.

[1.36] Wang, Yingxu, et al. "Perspectives on cognitive informatics and its future development: Summary of plenary panel II of IEEE ICCI'10." *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on*. IEEE, 2010.

[1.37] Wang, Yingxu. "In search of denotational mathematics: Novel mathematical means for contemporary intelligence, brain, and knowledge sciences." *Journal of Advanced Mathematics and Applications* 1.1 (2012): 4-26.

[1.38] Wang, Yingxu, et al. "A layered reference model of the brain (LRMB)."*Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 36.2 (2006): 124-133.

[1.39] Wang, Yingxu. "The OAR Model of Neural Informatics for Internal Knowledge Representation in the Brain." *IJCINI* 1.3 (2007): 66-77.

[1.40] Kennedy, David N. "Making connections in the connectome era."*Neuroinformatics* 8.2 (2010): 61-62.

[1.41] Sarivougioukas, John, Aristides Vagelatos, and Isaac Lagaris. "Use of Denotational Mathematics for the formal description of autonomous migration and polymorphism as prerequisites for mobility in home UbiHealth." *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015 IEEE 14th International Conference on*. IEEE, 2015.

[1.42] Von Lubitz, Dag, and Nilmini Wickramasinghe. "Networkcentric healthcare: applying the tools, techniques and strategies of knowledge management to create superior healthcare operations." *International journal of electronic healthcare* 2.4 (2006): 415-429.

[1.43] Belciug, Smaranda, and Florin Gorunescu. "Improving hospital bed occupancy and resource utilization through queuing modeling and evolutionary computation." *Journal of biomedical informatics* 53 (2015): 261-269.

[1.44] Piña, Ileana L., et al. "A framework for describing health care delivery organizations and systems." *American journal of public health* 105.4 (2015): 670-679.

[1.45] Pang, Zhibo, et al. "Design of a terminal solution for integration of in-home health care devices and services towards the Internet-of-Things." *Enterprise Information Systems* 9.1 (2015): 86-116.

[1.46] Leff, Bruce, et al. "Hospital at home: feasibility and outcomes of a program to provide hospital-level care at home for acutely ill older patients." *Annals of internal medicine* 143.11 (2005): 798-808.

[1.47] Chan, Marie, et al. "Smart homes—current features and future perspectives." *Maturitas* 64.2 (2009): 90-97.

Chspter-2

[2.1] dos Santos, Felipe, and Cristiano André da Costa. "UBIHEALTH: A UBIQUITOUS COLLABORATIVE SYSTEM TO SUPPORT TELEMEDICINE."

[2.2] Bali, Rajeev K., Ashish Dwivedi, and Raouf NG Naguib. "The efficacy of knowledge management for personalised healthcare." Studies in health technology and informatics 117 (2005): 104.

[2.3] Leff, Bruce. "Hospital at home." Geriatrics Models of Care. Springer International Publishing, 2015. 163-171.

[2.4] Von Lubitz, Dag, and Nilmini Wickramasinghe. "Networkcentric Healthcare: Outline of Entry Portal Concept." IJEBM 4.1 (2006): 16-28.

[2.5] Ferscha, A., Davies, N., Schmidt, A., & Streitz, N. (2011). Pervasive socio-technical fabric. Procedia Computer Science, 7, 88-91.

[2.6] Ruiz, Patricia, et al. "Introduction to Mobile Ad Hoc Networks." (2014).

[2.7] Hassanalieragh, M., Page, A., Soyata, T., Sharma, G., Aktas, M., Mateos, G., ... & Andreescu, S. (2015, June). Health Monitoring and Management Using Internet-of-Things (IoT) Sensing with Cloud-based Processing: Opportunities and Challenges. In Services Computing (SCC), 2015 IEEE International Conference on (pp. 285-292). IEEE.

[2.8] Stantchev, V., Barnawi, A., Ghulam, S., Schubert, J., & Tamm, G. (2015). Smart Items, Fog and Cloud Computing as Enablers of Servitization in Healthcare. Sensors & Transducers (1726-5479), 185(2).

[2.9] Callahan, Daniel. "The WHO definition of'health'." Hastings Center Studies(1973): 77-87.

[2.10] Mark Weiser. “The computer for the 21st century”, Scientific American, vol 265, no 3, pp 94-104, 1991. http://wiki.daimi.au.dk/pca/_files/weiser-orig.pdf

[2.11] Howard Rheingold, “Virtual Reality”. Summit Books. New York, NY. 1991.

[2.12] Mark Weiser. “Some computer science issues in ubiquitous computing”. CACM, vol 36, issue 7, 1993. http://dx.doi.org/10.1145/159544.159617

[2.13] Anind K. Day, “Understanding and Using Context”, Carnegie Mellon University Research Showcase @ CMU, Human-Computer Interaction Institute, School of Computer Science, 2001, Carnegie Mellon University, http://repository.cmu.edu/cgi/viewcontent.cgi?article=1033&context=hcii&sei-redir=1&referer=http%3A%2F%2Fscholar.google.gr%2Fscholar%3Fstart%3D40%26q%3Ddefinition%2Bubiquitous%2Bcomputing%26hl%3Del%26as_sdt%3D0%2C5%26as_vis%3D1#search=%22definition%20ubiquitous%20computing%22

[2.14] Karen Henricksen, Jadwiga Indulska, “Modelling and Using Imperfect Context Information”, 2005, The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through theAustralianFederal Government’s CRC Programme (Department of Industry, Science & Resources), http://pdf.aminer.org/000/538/624/modelling_and_using_imperfect_context_information.pdf

[2.15] Jacek Gwizdka, “What’s in the Context?”, A Position paper for CHI 2000 Workshop 11. Interactive Media Laboratory, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Canada, https://www.google.gr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&uact=8&ved=0CDwQFjAD&url=http%3A%2F%2Fwww.cc.gatech.edu%2Ffce%2Fcontexttoolkit%2Fchiws%2FJGwizdka.rtf&ei=JdWpVJzwO5HSaPnQgpAJ&usg=AFQjCNG1gVRA8o4x0PkJDxVufGVWguKbsw&bvm=bv.82001339,d.d2s

[2.16] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, Daniele Riboni, “A Survey of Context Modelling and Reasoning Techniques”, Pervasive and Mobile Computing, Elsevier, 2010, https://wwwold.cs.umd.edu/class/spring2014/cmsc818g/files/bettinisurvey.pdf

[2.17] Zimmermann Andreas, “A Context-Management Infrastructure for Ubiquitous Computing Environments”, The CONTEXT’05 Conference, Doctoral COnsortium took place at the Pierre & Marie Curie University, 2005, http://ftp.lip6.fr/lip6/reports/2005/lip6-2005-006.pdf

[2.18] Schilit Bill, “System architecture for context-aware mobile computing”. Columbia University (1995).

[2.19] Bazire Mary, Brezillon Patrick, “Understanding Context Before Using It”, A. Dey et al. (Eds.): CONTEXT 2005, LNAI 3554, pp. 29 – 40, 2005. Springer-Verlag Berlin Heidelberg 2005, http://www-sysdef.lip6.fr/~brezil/Pages2/Publications/CXT05-MB-PB(LNCS3554).pdf

[2.20] Matthias Baldauf, Schahram Dustdar, Florian Rosenberg, “A survey on context-aware systems”, Int. J. Ad Hoc and Ubiquitous Computing, Vol. 2, No. 4, 2007, http://www.cs.umd.edu/class/spring2014/cmsc818g/files/surveyoncontextawaresystems.pdf

[2.21] Satyanarayanan, Mahadev. "Pervasive computing: Vision and challenges."Personal Communications, IEEE 8.4 (2001): 10-17.

[2.22] Long S, Kooper R, Abowd GD, Atkeson CG. “Rapid prototyping of mobile contextaware applications: The Cyberguide case study”, Wireless Networks. 1997, vol 3, pages 421–433. http://dx.doi.org/10.1023/A:1019194325861

[2.23] Schilit WN., “System architecture for context-aware mobile computing”, [unpublished PhD thesis]: Columbia University, 1995.

[2.24] Musumba GW, Nyongesa HO, “Context awareness in mobile computing: A review”, Int J Machine Learn Appl. 2013;2(1), Art. #5, 10 pages, http://dx.doi.org/10.4102/ijmla.v2i1.5

[2.25] Leont’ev AN, “Activity, consciousness, and personality”, 1978, Prentice Hall.

[2.26] Engeström, Y.: Developmental studies of work as a testbench of activity theory. In Chaiklin, S. and J. Lave, (Eds.), “Understanding practice: perspectives on activity and context”, 1993, Cambridge: Cambridge University Press, pp. 64–103

[2.27] Li Yang, Landay A. James, “Exploring Activity-Based Ubiquitous Computing: Interaction Styles, Models and Tool Support”, In: Proceedings of CHI’06, Montreal, 2006, http://www.cs.tufts.edu/~jacob/workshop/papers/li.pdf

[2.28] Parisa Rashidi, Diane J. Cook, Lawrence B. Holder, Maureen Schmitter-Edgecombe, “Discovering Activities to Recognize and Track in a Smart Environment”, IEEE Trans Knowl Data Eng. 2011; 23(4): 527–539, doi: 10.1109/TKDE.2010.148

[2.29] Mckeevera Susan, Ye Juan, Coylec Lorcan, Bleakleya Chris, Dobson Simon, “Activity recognition using temporal evidence theory”, Journal of Ambient Intelligence and Smart Environments, 2(3): 253-269, 2010, http://dx.doi.org/10.3233/AIS-2010-0071

[2.30] Patkos Theodore, Chibani Abdelghani, Plexousakis Dimitris, Amirat Yacine. “A Production Rule-based Framework for Causal and Epistemic Reasoning”, in the 6th International Symposium on Rules (RuleML'12), pp. 120-135, 2012.

[2.31] Rabiner L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition" Proc. of the IEEE, 1989, Vol.77, No.2, pp.257--286.

[2.32] Patkos, Theodore, and Dimitris Plexousakis. "Efficient epistemic reasoning in partially observable dynamic domains using hidden causal dependencies."Proc. 9th NRAC (2011): 55-62.

[2.33] Aamodt Agnar, Plaza Enric, “Case-based reasoning: Foundational issues, methodological variations, and system approaches”, Artificial Intelligence Communications, 7(1):39–59, 1994.

[2.34] Ontañón Santiago, Mishra Kinshuk, Sugandh Neha, Ram Ashwin, “On-line Case-Based Planning”, in Computational Intelligence Journal, 2010, Volume 26, Issue 1, pp. 84-119. (Impact factor 5.378)

[2.35] Dempster AP., “A generalization of Bayesian inference (with discussion)”, J Roy Stat Soc, Series B 1968;30(2):205±47.

[2.36] Shafer G., “A mathematical theory of evidence”, Princeton: Princeton University Press, 1976.

[2.37] Sia C, Tonniges TF, Osterhus E, Taba S. “History of the medical home concept”, Pediatrics 2004;113(5 Suppl):1473– 8.

[2.38] Coughlin Teresa, Long Sharon, Sheen Edward, Tolbert Jennifer, “How Five Leading Safety-Net Hospitals Are Preparing For The Challenges And Opportunities Of Health Care Reform”, Health Affairs, 31, no.8 (2012):1690-1697, doi: 10.1377/hlthaff.2012.0258.

[2.39] Puig-Junoy Jaume, Casas Alejandro , Font-Planells Jaume, Escarrabill Joan, Hernandez Carme, Alonso Jordi, Farrero Eva, Vilagut Gemma, Roca Josep, “The impact of home hospitalization on healthcare costs of exacerbations in COPD patients”, Eur J Health Econ, 2008, DOI 10.1007/s10198-006-0029-y.

[2.40] Greaney Anna-Marie, O’Mathu´na Do´nal, Scott Anne, “Patient autonomy and choice in healthcare: self-testing devices as a case in point”, Med Health Care and Philos (2012) 15:383–395, DOI 10.1007/s11019-011-9356-6

[2.41] Preden J., Pahtma R., “Smart Dust Motes in Ubiquitous Computing Scenarios”, ELECTRONICS AND ELECTRICAL ENGINEERING, ISSN 1392 – 1215, T170 ELECTRONICS, 2009. No. 6(94).

[2.42] Yi-Luen Do Ellen, “Things that Think, Spaces that Sense, and Places that Play”, Design Computing & Human Centered Computing, College of Architecture & College of Computing Georgia Institute of Technology Atlanta, 2006, GA 30332-0155, USA, http://www.cc.gatech.edu/~ellendo

[2.43] Khan R., Khan S. U. S., Zaheer R., “Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges”, in Frontiers of Information Technology ({FIT)}, 2012 10th International Conference on, 2012, pp. 257–260.

[2.44] Jeong Sangjin, Youn Chan-Hyun, Shim Eun Bo, Kim Moonjung, Cho Young Min, Peng Limei, “An Integrated Healthcare System for Personalized Chronic Disease Care in Home–Hospital Environments”, IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE, VOL. 16, NO. 4, JULY 2012.

[2.45] Jeong Sangjin, Youn Chan-Hyun, Kim Yong-Woon, “A Chronic Disease Identification Scheme Using Radar Chart Method for Personalized Healthcare System”, Ubiquitous Information Technologies and Applications, Lecture Notes in Electrical Engineering 280, 13 DOI: 10.1007/978-3-642-41671-2_3, © Springer-Verlag Berlin Heidelberg 2014.

[2.46] Yuan Bingchuan, Herbert John, “Context-aware hybrid reasoning framework for pervasive healthcare”, Journal Personal and Ubiquitous Computing, Volume 18 Issue 4, April 2014, Pages 865-881.

[2.47] Hsieh Jui-chien, Hsu Meng-Wei, “A cloud computing based 12-lead ECG telemedicine service”, BMC Medical Informatics and Decision Making 2012, 12:77 http://www.biomedcentral.com/1472-6947/12/77

[2.48] Prantosh Kumar Paul, “VIRTUAL WORLD: EMPOWERED BY CLOUD COMPUTING- A CONCEPTUAL STUDY”, International Monthly Refereed Journal of Research In Management & Technology, 82, ISSN – 2320-0073 Volume II, November’13.

[2.49] Preuveneers, D., Berbers, Y. (2008), “Encoding Semantic Awareness in Resource-Constrained Devices,” IEEE Intelligent Systems, 23(2), pp.26-33, March (2008).

[2.50] Gold Marthe, Stevenson David, Fryback Dennis,”HALYS AND QALYS AND DALYS, OH MY:Similarities and Differences in Summary Measures of Population Health”, Annu. Rev. Public Health 2002. 23:115–34, 2002, DOI: 10.1146/annurev.publhealth.23.100901.140513

[2.51] Bohn Jürgen, Gärtner Felix, Vogt Harald, “Dependability Issues of Pervasive Computing in a Healthcare Environment”, book title: Security in Pervasive Computing, Lecture Notes in Computer Science, Volume 2802, 2004, pp 53-70, Springer Berlin Heidelberg, doi:10.1007/978-3-540-39881-3_8.

[2.52] Tegegne Tesfa, Kanagwa Benjamin, van der Weide Theo, “Service Discovery Framework for Personalized eHealth”, The 2010 International Conference on Services Science, Management and Engineering (SSME 2010) Tianjin, China, December 26-28 2010 Tianjin, IEEE Press ISBN 978-0-7695-4128-0.

[2.53] Paganelli Federica, Giuli Dino, “An Ontology-Based System for Context-Aware and Configurable Services to Support Home-Based Continuous Care”, IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE, VOL. 15, NO. 2, MARCH 2011.

[2.54] Koutkias V.G., Chouvarda I., Triantafyllidis A., Malousi A., Giaglis G.D., Maglaveras N., “A Personalized Framework for Medication Treatment Management in Chronic Care”, Information Technology in Biomedicine, IEEE Transactions on, 2010, Volume:14 , Issue: 2, DOI: 10.1109/TITB.2009.2036367

[2.55] Cayirli Tugba, Veral Emre, Rosen Harry, “Assessment of patient classification in appointment system design”, Production and Operations Management, 2008, Vol. 17, No 3, pp 338-353, DOI: 10.3401/poms.1080.0031.

[2.56] Busnel Pierre, Giroux Sylvain, “Security, Privacy, and Dependability in Smart Homes: A Pattern Catalog Approach”, Book: Aging Friendly Technology for Health and Independence, 2010, pp 24-31, DOI: 10.1007/978-3-642-13778-5_4.

[2.57] Henricksen Karen, Indulska Jadwiga, Rakotonirainy Andry, “Using context and preferences to implement self-adapting pervasive computing applications”, Software. Practice and Experience, 2006; 36:1307–1330, Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/spe.760

[2.58] Rodríguez Marcela, Jesus Favela,. "A Framework for supporting autonomous agents in ubiquitous computing environments", Proceedings of System Support for Ubiquitous Computing Workshop at the Fifth Annual Conference on Ubiquitous Computing (UbiComp 2003), Seattle, Washington. 2003.

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

[2.60] Dey Anind, "Understanding and using context", Personal and ubiquitous computing 5.1 (2001): 4-7.

[2.61] Javidaneh Ali, Farid Karimipour, "Exploring the Addressing Contexts", Context-Awareness in Geographic Information Services (CAGIS 2014) (2014): 29.

[2.62] Alnanih R., Radhakrishnan T., Ormandjieva O., “Characterising Context for Mobile User Interfaces in Health Care Applications”, Procedia Computer Science, Volume 10, 2012, Pages 1086–1093, ANT 2012 and MobiWIS 2012, doi:10.1016/j.procs.2012.06.153

[2.63] Koichi Egami, Matsumoto Shinsuke, Nakamura Masahide, "Ubiquitous cloud: Managing service resources for adaptive ubiquitous computing", Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on. IEEE, 2011.

[2.64] Golm Michael, Jürgen Kleinöder, "Ubiquitous computing and the need for a new operating system architecture", Online at http://www4. informatik. uni-erlangen. de/Projects/JX/Papers/ubitools01. pdf (2001).

[2.65] Nandal S. K., Chaba Y., “Location sensitive service discovery protocol for mobility support in ubiquitous computing environment”, International Journal of Computer Engineering & Science, Jan. 2014. Volume 4, Issue 1, pp. 25-30, ©IJCES, ISSN: 22316590.

[2.66] Zaplata Sonja, Kottke Kristian, Meiners Matthias, Lamersdorf Winfried, "Towards runtime migration of WS-BPEL processes", Service-Oriented Computing, ICSOC / Service Wave 2009 Workshops,. Springer Berlin Heidelberg, 2010.

[2.67] Kunze P. Christian,” DEMAC: A Distributed Environment for Mobility Aware Computing”, University of Hamburg, Department of Informatics, Vogt-K¨olln-Straße 30, 22527 Hamburg, Germany, kunze@informatik.uni-hamburg.de, http://vsis-www.informatik.uni-hamburg.de.

[2.68] Munnelly Jennifer, Siobhán Clarke, "Infrastructure for ubiquitous computing: Improving quality with modularisation", Proceedings of the 2008 AOSD workshop on Aspects, components, and patterns for infrastructure software. ACM, 2008.

[2.69] Kortuem Gerd, Schneider Jay, Preuitt Dustin, Thompson Thaddeus, Fickas Stephen, Segall Zary, "When peer-to-peer comes face-to-face: Collaborative peer-to-peer computing in mobile ad-hoc networks." Peer-to-Peer Computing, 2001. Proceedings. First International Conference on. IEEE, 2001.

[2.70] Pathak Parth, Rudra, Dutta, "A survey of network design problems and joint design approaches in wireless mesh networks" Communications Surveys & Tutorials, IEEE 13.3 (2011): 396-428.

[2.71] Erfani1 Seyed Hossein, Javadi H. S. Hamid, Rahmani Amir Masoud, “Analysis of Key Management Schemes in Dynamic Wireless Sensor Networks”, ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 1, No.13 , January 2015, ISSN : 2322-5157.

[2.72] Kiciman Emre, “Using Internet Services to Manage Massive Evolving Information for Ubiquitous Computing Systems”, In: ADJUNCT PROCEEDINGS. 2002. p. 71.

[2.73] Sanaei Z., Abolfazli S., Gani A., Khokhar, R. H., “Tripod of requirements in horizontal heterogeneous mobile cloud computing”, 2012, arXiv preprint arXiv:1205.3247.

[2.74] Sanaei Z., Abolfazli S., Gani A., Buyya, R., “Heterogeneity in mobile cloud computing: taxonomy and open challenges”, Communications Surveys & Tutorials, IEEE, 2014, 16(1), 369-392.

[2.75] Bashir Rab Nawaz, Qadri Salman, Saleem Rana Muhammad, Naeem Muhammad, Ghafoor Yasir, “Human Computer Interaction (HCI) in Ubiquitous Computing”, International Journal of Innovation and Applied Studies, ISSN 2028-9324 Vol. 9 No. 2 Nov. 2014, pp. 534-540, © 2014 Innovative Space of Scientific Research Journals

[2.76] Karray Fakhreddine, Alemzadeh Milad, Saleh Jamil Abou, Arab Mo Nours, “Human-Computer Interaction: Overview on State of the Art”, INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS, VOL. 1, NO. 1, MARCH 2008.

[2.77] Poslad Stefan, “Ubiquitous Computing: Smart Devices, Environments and Interactions”, book, © 2009 John Wiley & Sons, Ltd. ISBN: 978-0-470-03560-3.

[2.78] Van den Broek Egon L., “Affective Signal Processing (ASP): Unraveling the mystery of emotions”, University of Twente, Ph.D. Thesis, 2011.

[2.79] Van den Broek Egon L., "Ubiquitous emotion-aware computing", Personal and Ubiquitous Computing, 17.1 (2013): 53-67.

[2.80] Atzori Luigi, Iera Antonio, Morabito Giacomo, "The internet of things: A survey", Computer networks, 54.15, 2010, 2787-2805, doi:10.1016/j.comnet.2010.05.010.

[2.81] Friedewald Michael, Raabe Oliver, "Ubiquitous computing: An overview of technology impacts", Telematics and Informatics 28.2, 2011, 55-65.

[2.82] Fuentes, A., Bolívar, E., Tagarro, J., & Gutiérrez, J. M., “Integration of Ubiquitous Computing Technologies: a Practical Case”, In ICUC 2006.

[2.83] Strang, Thomas, Claudia Linnhoff-Popien, "Service interoperability on context level in ubiquitous computing environments", Intl. Conf. on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet (SSGRR2003w). 2003.

[2.84] Georgantas Nikolaos, Issarny Valerie, Mokhtar Sonia Ben, Bromberg Yerom-David, Bianco Sebastien, Thomson Graham, Raverdy Pierre-Guillaume, Urbieta Aitor, Cardoso Roberto Speicys, "Middleware architecture for ambient intelligence in the networked home", In Handbook of Ambient Intelligence and Smart Environments, 2010 (pp. 1139-1169). Springer US.

[2.85] Davidyuk, O., Georgantas, N., Issarny, V., Riekki, J., "MEDUSA: Middleware for end-user composition of ubiquitous applications", Handbook of research on ambient intelligence and smart environments: trends and perspectives, 2011, 11, 197-219.

[2.86] Flores, Andrés P., Macario Polo Usaola, "Considerations upon interoperability on pervasive computing environments", VI Workshop de Investigadores en Ciencias de la Computación. 2004.

[2.87] Sanaei Zohreh, Abolfazli Saeid, Khodadadi Touraj, Xia Feng, "Hybrid Pervasive Mobile Cloud Computing: Toward Enhancing Invisibility", INFORMATION-An international interdisciplinary journal, 2013, 16(11),pp.1–12.

[2.88] Marquardt, N., Gross, T., Carpendale, S. Greenberg, S., "Revealing the Invisible: Visualising the Location and Event Flow of Distributed Physical Devices", In Proceedings of the Fourth International Conference on Tangible Embedded Interaction - TEI 2010 (Jan. 24-27, Cambridge, MA). ACM, N.Y., 2010. pp. 41-48. (ISBN: 978-1-60558-841-4).

[2.89] Heer Jeffrey, Khooshabeh Peter, "Seeing the invisible." Workshop on Invisible and Transparent Interfaces at AVI. 2004.

[2.90] Davidyuk, O., Georgantas, N., Issarny, V., & Riekki, J., "MEDUSA: Middleware for end-user composition of ubiquitous applications", Handbook of research on ambient intelligence and smart environments: trends and perspectives, 11, 2011, 197-219.

[2.91] Vladoiu, M., Cassens, J., Constantinescu, Z., "FACE–a Knowledge-Intensive Case-Based Architecture for Context-Aware Services", In Networked Digital Technologies (pp. 533-544), 2010, Springer Berlin Heidelberg.

[2.92] Goumopoulos, C., Kameas, A., "An Ontology-Driven Approach and a Context Management Framework for Ubiquitous Computing Applications", InTheory and Applications of Ontology: Computer Applications (pp. 463-485), 2010, Springer Netherlands.

[2.93] Lee, S., Gon Kim, B., Kim, H., "An integrated view of knowledge management for performance", 2012, Journal of Knowledge management, 16(2), 183-203.

[2.94] Trevor, J., Hubert, D. M., Schilit, B. N., "Issues in personalizing shared ubiquitous devices", In UbiComp 2002: Ubiquitous Computing (pp. 56-72), 2002, Springer Berlin Heidelberg.

[2.95] Kawsar, F., Fujinami, K., Nakajima, T., Park, J. H., Yeo, S. S., "A portable toolkit for supporting end-user personalization and control in context-aware applications", Multimedia Tools and Applications, 47(3), 2010, 409-432.

[2.96] Sharma M.. "A Survey of Recommender Systems: Approaches and Limitations", International Journal of Innovations in Engineering and Technology, 2013, ICAECE.

[2.97] Thein, T., Chi, S. D., Park, J. S., "Increasing availability and survivability of cluster head in WSN", In Grid and Pervasive Computing Workshops, 2008. GPC Workshops' 08. The 3rd International Conference on (pp. 281-285), 2008, IEEE.

[2.98] Silas, S., Ezra, K., Rajsingh, E. B., "A novel fault tolerant service selection framework for pervasive computing", Human-centric Computing and Information Sciences, 2012, 2(1), 1-14.

[2.99] Wason Ritika, Ahmed P., Qasim Rafiq M., "Automata-Based Software Reliability Paradigm for Ubiquitous Systems", International Journal of Computer Science & Engineering Technology (IJCSET), Vol. 4 No. 07 Jul 2013, ISSN : 2229-3345.

[2.100] Youssef A., "Towards pervasive computing environments with cloud services", International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC), 2013, 4(3), 1-9.

[2.101] Schmietendorf A., Wille C., Fiegler A., Dumke R. R., Neumann, R., "COSMIC Functional Size Measurement of Cloud Systems", InSoftware Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2013 Joint Conference of the 23rd International Workshop on (pp. 33-37), October 2013, IEEE.

[2.102] Ali H. A., Salem M. M., & Hamza A. A., "A framework for scalable autonomous p2p resource discovery for the grid implementation", International Journal of Computer Systems Science & Engineering, 27(4), 2012, 275-287.

[2.103] Jalil K. A., Rahman Q. B. A., “Authenticating Devices in Ubiquitous Computing Environment”, International Journal of Cyber-Security and Digital Forensics (IJCSDF), 2012, 1(2), 75-81.

[2.104] Yao L., Wang L., Kong X., Wu G., Xia F., “An inter-domain authentication scheme for pervasive computing environment”, Computers & Mathematics with Applications, 2010, 60(2), 234-244.

[2.105] Agrafioti F., Bui F. M., Hatzinakos D., “On supporting anonymity in a BAN biometric framework”, In Digital Signal Processing, 2009 16th International Conference on , July 2009, (pp. 1-6). IEEE.

[2.106] Venkatasubramanian K. K., Banerjee A., Gupta, S. K. S., “PSKA: usable and secure key agreement scheme for body area networks”, Information Technology in Biomedicine, IEEE Transactions on, 2010, 14(1), 60-68.

[2.107] Lim S., Oh T. H., Choi Y. B., Lakshman T., “Security issues on wireless body area network for remote healthcare monitoring”, In Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2010 IEEE International Conference on (pp. 327-332). IEEE.

[2.108] Theoharidou M., Tsalis N., Gritzalis, D., “Smart Home Solutions for Healthcare: Privacy in Ubiquitous Computing Infrastructures”, Handbook of Smart Homes, Health Care and Well-Being, (2014).

[2.109] Davies N., Friday A., Clinch S., Sas C., Langheinrich M., Ward G., Schmidt A., “Security and Privacy Implications of Pervasive Memory Augmentation”, Pervasive Computing, IEEE, 2015, 14(1).

[2.110] Mahmood M. A., Seah W. K., “Event reliability in wireless sensor networks. In Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)”, 2011 Seventh International Conference on (pp. 377-382). IEEE.

[2.111] Matin M. A., Islam M. M. “Overview of Wireless Sensor Network”, INTECH, 2012, Open Access Publisher.

[2.112] Bakillah M., Liang S. H., Zipf A., Mostafavi M. A., “A dynamic and context-aware semantic mediation service for discovering and fusion of heterogeneous sensor data”, Journal of Spatial Information Science, 2015, (6), 155-185.

[2.113] Izadi D., Abawajy J. H., Ghanavati S., Herawan T., “A Data Fusion Method in Wireless Sensor Networks”, Sensors, 2015, 15(2), 2964-2979.

[2.114] Bohn J., Coroamă V., Langheinrich M., Mattern F., Rohs M., “Social, economic, and ethical implications of ambient intelligence and ubiquitous computing“, In Ambient intelligence (pp. 5-29), 2005, Springer Berlin Heidelberg.

[2.115] Augusto J. C., Nakashima H., Aghajan, H., “Ambient intelligence and smart environments: A state of the art”, In Handbook of ambient intelligence and smart environments (pp. 3-31), 2010, Springer US.

[2.116] Meulendijk M., van de Wijngaert L., Brinkkemper S., Leenstra, H., “AmI in good care?: Developing design principles for ambient intelligent domotics for elderly”, Informatics for Health and Social Care , 2011, 36 (2), 75-88.

[2.117] Julia Clark, McGee-Lennon Marilyn, "A stakeholder-centred exploration of the current barriers to the uptake of home care technology in the UK", Journal of Assistive Technologies 5.1 (2011): 12-25.

[2.118] Ward Jeremy, Gordon Jill, Field Michael, Lehmann Harold, “Communication and information technology in medical education”, MEDICAL EDUCATION QUARTET, THE LANCET • Vol 357 • March 10, 2001.

[2.119] Graaf, B, “Model-Driven Evolution of Software Architectures”, Ph.D Dissertation: Technical University of Delft, 2007.

[2.120] de Boer R. C., Lago P., Telea A., van Vliet H., “Ontology-driven visualization of architectural design decisions”, Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture (WICSA/ECSA), 2009.

[2.121] Chomsky Noam, “On Certain Formal Properties of Grammars”, Information and Control 2 (2): 137–167, 1959, doi:10.1016/S0019-9958(59)90362-6.

[2.122] Zhang Xiuguo, Liu Hongbo, Abraham Ajith, “A Novel Process Network Model for Interacting Context-Aware Web Services”, IEEE TRANSACTIONS ON SERVICES COMPUTING, 2012, DOI: 10.1109/TSC.2012.6.

[2.123] Busse, Reinhard, and Annette Riesberg. "Health care systems in transition: Germany." (2004).

[2.124] Starfield, Barbara, and George G. Reader. "Primary Care: Concept, Evaluation, and Policy." Journal of Public Health Policy 14.4 (1993): 500-500.

[2.125] Berendsen, Brenda AJ, et al. "The implementation and sustainability of a combined lifestyle intervention in primary care: mixed method process evaluation." BMC family practice 16.1 (2015): 37.

[2.126] Spicker, Paul. "Poverty and social security: concepts and principles." (2013).

[2.127] Beckfield, Jason, Sigrun Olafsdottir, and Benjamin Sosnaud. "Healthcare systems in comparative perspective: Classification, convergence, institutions, inequalities, and five missed turns." Annual review of sociology 39 (2013): 127-146.

[2.128] de Oliveira, Francisco Eduardo Barreto. Basic issues in reforming social security systems. No. 0071. 2015.

[2.129] Palfrey, J. S., Sofis, L. A., Davidson, E. J., Liu, J., Freeman, L., & Ganz, M. L. (2004). The Pediatric Alliance for Coordinated Care: evaluation of a medical home model. Pediatrics, 113(Supplement 4), 1507-1516.

[2.130] Schiefer, Michael. "Smart Home Definition and Security Threats." IT Security Incident Management & IT Forensics (IMF), 2015 Ninth International Conference on. IEEE, 2015.

[2.131] Nazir, Mohsin, et al. "Cloud Computing: Current Research Challenges." Book Chapter of Cloud Computing: Reviews, Surveys, Tools, Techniques and Applications-An Open-Access eBook published by HCTL Open (2015).

[2.132] Coady, Yvonne, et al. "Distributed Cloud Computing: Applications, Status Quo, and Challenges." ACM SIGCOMM Computer Communication Review 45.2 (2015): 38-43.

[2.133] Oriwoh, Edewede, and Marc Conrad. "'Things' in the Internet of Things: Towards a Definition." International Journal of Internet of Things 4.1 (2015): 1-5.

[2.134] Want, Roy, Bill N. Schilit, and Scott Jenson. "Enabling the Internet of Things."Computer 1 (2015): 28-35.

[2.135] Byrne, David, et al. "Long-term care, formal home health care, and informal care." mimeographed, University of Virginia (2002).

[2.136] Anderson, Wayne L., and Joshua M. Wiener. "The Impact of Assistive Technologies on Formal and Informal Home Care." The Gerontologist 55.3 (2015): 422-433.

[2.137] Van Houtven, Courtney Harold, and Edward C. Norton. "Informal care and health care use of older adults." Journal of health economics 23.6 (2004): 1159-1180.

[2.138] Hwang, AmyS, Khai N. Truong, and Alex Mihailidis. "Using participatory design to determine the needs of informal caregivers for smart home user interfaces."Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on. IEEE, 2012.

[2.139] Bolin, Kristian, Bjorn Lindgren, and Petter Lundborg. "Informal and formal care among single-living elderly in Europe." Health economics 17.3 (2008): 393-409.

[2.140] Reese, Peter P., et al. "Hospitalist home visit program identifies medication errors in a vulnerable general medicine population." JCOM 10.4 (2003).

[2.141] Porter, Michael E., Erika A. Pabo, and Thomas H. Lee. "Redesigning primary care: a strategic vision to improve value by organizing around patients' needs."Health Affairs 32.3 (2013): 516-525.

[2.142] Porter, Michael E., Erika A. Pabo, and Thomas H. Lee. "Redesigning primary care: a strategic vision to improve value by organizing around patients' needs."Health Affairs 32.3 (2013): 516-525.

[2.143] Paul III, David P. "An Innovation in Healthcare Delivery: Hospital at Home."Journal of Management Policy and Practice 14.6 (2013): 73-91.

[2.144] Tackett, John J., et al. "Comprehensive Home-Based Acute Care Medical Program Initiative for Older Noncritical Surgical Patients." JAMA surgery149.11 (2014): 1212-1213.

[2.145] Kadu, Mudathira K., and Paul Stolee. "Facilitators and barriers of implementing the chronic care model in primary care: a systematic review." BMC family practice 16.1 (2015): 12.

[2.146] Boult, Chad, and Jennifer L. Wolff. "“Guided Care” for People with Complex Health Care Needs." Geriatrics Models of Care. Springer International Publishing, 2015. 139-145.

[2.147] Ottmann, Goetz, Jacqui Allen, and Peter Feldman. "A systematic narrative review of consumer-directed care for older people: implications for model development." Health & social care in the community 21.6 (2013): 563-581.

[2.148] Yue, Jirong, Tammy T. Hshieh, and Sharon K. Inouye. "Hospital Elder Life Program (HELP)." Geriatrics Models of Care. Springer International Publishing, 2015. 25-37.

[2.149] Fox, Mary T., et al. "Acute care for elders components of acute geriatric unit care: systematic descriptive review." Journal of the American Geriatrics Society 61.6 (2013): 939-946.

[2.150] Chapin, R. K., Wendel, C., Lee, R., Landry, S., Zimmerman, M. K., Oslund, P., ... & Sergeant, J. (2013). Program of All-inclusive Care for the Elderly (PACE) Medicaid Cost-Benefit Study.

[2.151] Chochinov, Harvey Max. "Dignity-conserving care—a new model for palliative care: helping the patient feel valued." JaMa 287.17 (2002): 2253-2260.

[2.152] Comart, Jody, et al. "Palliative care for long-term care residents: Effect on clinical outcomes." The Gerontologist 53.5 (2013): 874-880.

[2.153] Brock, Jane, et al. "Association between quality improvement for care transitions in communities and rehospitalizations among Medicare beneficiaries." Jama 309.4 (2013): 381-391.

[2.154] Cool, Julie. Canada's Aging Population and Public Policy: The Effects on Home Care. Library of Parliament, 2012.

[2.155] Howrey, Bret T., et al. "Partnering with patients, families, and communities."Fam Med 47.8 (2015): 604-11.

[2.156] Russo, Valentina, Mario Ciampi, and Massimo Esposito. "A Business Process Model for Integrated Home Care." Procedia Computer Science 63 (2015): 300-307.

[2.157] Arbaoui, Selma, Nathalie Cislo, and Natalie Smith-Guerin. "Home healthcare process: Challenges and open issues." arXiv preprint arXiv:1206.5430 (2012).

[2.158] Hamek, Saliha, et al. "Cooperation in healthcare-theoretical and methodological issues: a study of two situations: hospital and home care." Proceedings of the 2005 annual conference on European association of cognitive ergonomics. University of Athens, 2005.

[2.159] Lanzarone, Ettore, Andrea Matta, and Evren Sahin. "Operations management applied to home care services: the problem of assigning human resources to patients." Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 42.6 (2012): 1346-1363.

[2.160] Weir, Charlene R., et al. "A qualitative evaluation of the crucial attributes of contextual Information necessary in EHR design to support patient-centered medical home care." BMC medical informatics and decision making 15.1 (2015): 30.

[2.161] Quadri, Imran Rafiq, et al. "Expressing embedded systems configurations at high abstraction levels with UML MARTE profile: Advantages, limitations and alternatives." Journal of systems architecture 58.5 (2012): 178-194.

[2.162] Matta, Andrea, et al. "Modelling home care organisations from an operations management perspective." Flexible Services and Manufacturing Journal 26.3 (2014): 295-319.

[2.163] Privat, Gilles, Mengxuan Zhao, and Laurent Lemke. "Towards a Shared Software Infrastructure for Smart Homes, Smart Buildings and Smart Cities."International Workshop on Emerging Trends in the Engineering of Cyber-Physical Systems, Berlin. 2014.

[2.164] Zeigler, B. P., et al. "Care Coordination: Formalization of Pathways for Standardization and Certification." (2014).

[2.165] Ing, William JZ, and Ron Summers. "Technological Design Considerations for Home-based Health care." Measurement and Control 48.6 (2015): 183-188.

[2.166] Wang, Yingxu. "On concept algebra: A denotational mathematical structure for knowledge and software modeling." International Journal of Cognitive Informaticsand Natural Intelligence 2.2 (2008): 1.

[2.167] Wang, Yingxu. "On system algebra: A denotational mathematical structure for abstract system modeling." International Journal of Cognitive Informaticsand Natural Intelligence 2.2 (2008): 20.

[2.168] Wang, Y. (2008). On contemporary denotational mathematics for computational intelligence. In Transactions on computational science II (pp. 6-29). Springer Berlin Heidelberg.

[2.169] Wang, Yingxu. "In search of denotational mathematics: Novel mathematical means for contemporary intelligence, brain, and knowledge sciences." Journal of Advanced Mathematics and Applications 1.1 (2012): 4-26.

Chspter-3

[3.1] Wang, Y. (2008). On contemporary denotational mathematics for computational intelligence. In Transactions on computational science II (pp. 6-29). Springer Berlin Heidelberg.

[3.2] Wang, Yingxu. "In search of denotational mathematics: Novel mathematical means for contemporary intelligence, brain, and knowledge sciences." Journal of Advanced Mathematics and Applications 1.1 (2012): 4-26.

[3.3] WHO (1946). WHO Definition of Health. Preamble to the Constitution of the World Health Organization. International Health Conference NY, 19-22 June, 1946, Official Records of the World Health Organization.

[3.4] King, C. H. (2015). Health metrics for helminth infections. Acta tropica, 141, 150-160.

[3.5] Evans, R. G., & Stoddart, G. L. (2003). Consuming research, producing policy?. American Journal of Public Health, 93(3), 371-379.

[3.6] Strang, T., Linnhoff-Popien, C., & Frank, K. (2003, January). CoOL: A context ontology language to enable contextual interoperability. In Distributed applications and interoperable systems (pp. 236-247). Springer Berlin Heidelberg.

[3.7] Kosek, A. M., Syed, A. A., & Kerridgey, J. M. (2010, June). RDF recipes for context-aware interoperability in pervasive systems. In Computers and Communications (ISCC), 2010 IEEE Symposium on (pp. 1017-1022). IEEE.

[3.8] Siolas, G., Caridakis, G., Mylonas, P., Kollias, S., & Stafylopatis, A. (2013, December). Context-Aware User Modeling and Semantic Interoperability in Smart Home Environments. In Semantic and Social Media Adaptation and Personalization (SMAP), 2013 8th International Workshop on (pp. 27-32). IEEE.

[3.9] Wan, K. (2009). A brief history of context. arXiv preprint arXiv:0912.1838.

Chspter-4

[4.1] Wang, Y., Ngolah, C. F., Zeng, G., Sheu, P. C., Choy, C. P., & Tian, Y. (2010). The Formal Design Model of a Real-Time Operating System (RTOS+):

Conceptual and Architectural Frameworks. International Journal of Software Science and Computational Intelligence (IJSSCI), 2(2), 105-122.

[4.2] Chen, Chi-Tsong. Linear system theory and design. Oxford University Press, Inc., 1995.

[4.3] Wang, Y., Tan, X., & Ngolah, C. F. (2012). Design and Implementation of an Autonomic Code. Breakthroughs in Software Science and Computational Intelligence, 196.

[4.4] Wang, Y. (2008). RTPA: A denotational mathematics for manipulating intelligent and computational behaviors. International Journal of Cognitive Informatics and Natural Intelligence (IJCINI), 2(2), 44-62.

[4.5] Wang, Y. (2010). Cognitive robots. Robotics & Automation Magazine, IEEE, 17(4), 54-62.

[4.6] Wang, Y., Tian, Y., & Hu, K. (2012). Semantic manipulations and formal ontology for machine learning based on concept algebra. Cognitive Informatics for Revealing Human Cognition: Knowledge Manipulations in Natural Intelligence: Knowledge Manipulations in Natural Intelligence, 194.

[4.7] Y. Wang, Cyprian F. Ngolah, G. Zeng, C.-Y. Sheu, C. Philip Choy, “The Formal Design Model of a Real-Time Operating System (RTOS+):Conceptual and Architectural Frameworks”, International Journal of Software Science and Computational Intelligence, 2(2), 105-122, April-June 2010.

[4.8] Y. Wang, G. Zeng, C. F. Ngolah, Philip C.-Y. Sheu, Y. Tian “The Formal Design Model of a Real-Time Operating System (RTOS+): Static and Dynamic Behaviors”, International Journal of Software Science and Computational Intelligence, 2(3), 79-105, July-September 2010.

[4.9] Y. Wang, “The Cognitive Processes of Formal Inferences”, Int’l Journal of Cognitive Informatics and Natural Intelligence, 1(4), 75-86, October-December 2007

[4.10] Wang, Y., “On concept algebra for computing with words (CWW)”, International Journal of Semantic Computing, 4(03), 331-356, (2010).

[4.11] Tian, Y., Wang, Y., Gavrilova, M. L., & Ruhe, G. (2011). A formal knowledge representation system (FKRS) for the intelligent knowledge base of a cognitive learning engine. International Journal of Software Science and Computational Intelligence (IJSSCI), 3, 1-17.

[4.12] Wang Y., “The Cognitive Processes of Formal Inferences”, Int’l Journal of Cognitive Informatics and Natural Intelligence, 1(4), 75-86, October-December 2007

[4.13] Wang Y., “A Cognitive Informatics Reference Model of Autonomous Agent Systems (AAS)”, Int’l Journal of Cognitive Informatics and Natural Intelligence, 3(1), 1-16, January-March 2009.

Chspter-5

[5.1] Sousa, João Pedro, and David Garlan. "Aura: an architectural framework for user mobility in ubiquitous computing environments." Software Architecture. Springer US, 2002. 29-43.

[5.2] Wang, Z., & Garlan, D. (2000). Task-driven computing (No. CMU-CS-00-154). CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE.

[5.3] Mokhtar, S. B., Georgantas, N., & Issarny, V. (2005, January). Ad hoc composition of user tasks in pervasive computing environments. In Software Composition (pp. 31-46). Springer Berlin Heidelberg.

[5.4] Mokhtar, S. B., Preuveneers, D., Georgantas, N., Issarny, V., & Berbers, Y. (2008). EASY: Efficient semAntic Service discoverY in pervasive computing environments with QoS and context support. Journal of Systems and Software, 81(5), 785-808.

[5.5] Pan, G., Xu, Y., Wu, Z., Li, S., Yang, L. T., Lin, M., & Liu, Z. (2011). TaskShadow: toward seamless task migration across smart environments. IEEE Intelligent Systems, 26(3), 50-57.

[5.6] Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H., & Nahrstedt, K. (2002). Gaia: a middleware platform for active spaces. ACM SIGMOBILE Mobile Computing and Communications Review, 6(4), 65-67.

[5.7] Cerqueira, R., Hess, C. K., Román, M., & Campbell, R. H. (2001, September). Gaia: A development infrastructure for active spaces. In Workshop on Application Models and Programming Tools for Ubiquitous Computing (held in conjunction with the UBICOMP 2001) (Vol. 2, p. 11).

[5.8] Preuveneers, D., & Berbers, Y. (2008). Pervasive services on the move: Smart service diffusion on the OSGi framework. In Ubiquitous Intelligence and Computing (pp. 46-60). Springer Berlin Heidelberg.

[5.9] Preuveneers, D., & Berbers, Y. (2010). Context-driven migration and diffusion of pervasive services on the OSGi framework. Int. J. Autonomous and Adaptive Communications Systems, Vol. 3, No. 1, 2010.

[5.10] Gilbert, B., Goode, A., & Satyanarayanan, M. (2010). Pocket ISR: Virtual machines anywhere. Tech. Rep. CMU-CS-10-112, School of Computer Science, Carnegie Mellon University.

[5.11] Satyanarayanan, M., Kozuch, M. A., Helfrich, C. J., & O'Hallaron, D. R. (2005). Towards seamless mobility on pervasive hardware. Pervasive and Mobile Computing, 1(2), 157-189.

[5.12] Grimm, R. (2004). One. world: Experiences with a pervasive computing architecture. IEEE Pervasive Computing, 3(3), 22-30.

[5.13] Grimm R., Davis J., Lemar E., MacBeth A., Swanson S., Gribble S., Anderson T., Bershad B., Borriello G., Wetherall D. (2001, September). System-level programming abstractions for ubiquitous computing. In Proc. of the 8th HotOS.

[5.14] Pan, G., Xu, Y., Wu, Z., Li, S., Yang, L. T., Lin, M., & Liu, Z. (2011). TaskShadow: toward seamless task migration across smart environments. IEEE Intelligent Systems, 26(3), 50-57.

[5.15] Wu, Z., & Pan, G. (2013). Task Migration in SmartShadow. In SmartShadow: Models and Methods for Pervasive Computing (pp. 25-42). Springer Berlin Heidelberg

[5.16] Paz-Lopez, A., Varela, G., Monroy, J., Vazquez-Rodriguez, S., & Duro, R. J. (2008). HI3 Project: General purpose ambient intelligence architecture. Proc. AITAmI, 77-81.

[5.17] Paz-Lopez, A., Varela, G., Becerra, J. A., Vazquez-Rodriguez, S., & Duro, R. J. (2013). Towards ubiquity in ambient intelligence: User-guided component mobility in the HI 3 architecture. Science of Computer Programming, 78(10), 1971-1986.

[5.18] Strang, T., Linnhoff-Popien, C., & Frank, K. (2003, January). CoOL: A context ontology language to enable contextual interoperability. In Distributed applications and interoperable systems (pp. 236-247). Springer Berlin Heidelberg.

APPENDIX

1. Author’s Publications

2. Author’s Resume

AUTHOR’S PUBLICATIONS

1) J. Sarivougioukas, I. Lagaris and A. Vagelatos, “Time in the Ubiquitous Computing Context for Home Healthcare applications”, Ubiquitous Computing and Communication Journal, Vol. 8, No 2, 2013.
2) J. Sarivougioukas and A. Vagelatos: Analyzing Medical Contexts in Ubiquitous Computing Home Environments with Denotational Mathematics, 13th IEEE International Conference on Cognitive Informatics & Cognitive Computing, Aug. 2014, London UK.
3) John Constantine Sarivougioukas, Aristides Vagelatos, “Open Data: A condition for the Clinical Governance“, 7th Pan Hellenic & 6th Pan European Scientific & Professional Nursing Conference, Ioannina, Greece, May 8-11, 2014, Hellenic Nursing Association, wwe.ene.gr.
4) J. Sarivougioukas, A. Vagelatos, “RFID at the hospital: Application in clothing”, 15o Pan-Hellenic Nursing Conference, December 12-14, 2014, Meteora, Greece.
5) J. Sarivougioukas, D. Nikolaou, S. Kalatzi, “Design Specifications of an integrated hospital system for Anti-microbiological Copper” 15o Pan-Hellenic Nursing Conference, December 12-14, 2014, Meteora, Greece.
6) J. Sarivougioukas, A. Vagelatos and I. Lagaris: Administration of Medical Contexts with Denotational Mathematics in Ubiquitous Computing Home Environments, International Journal of Software Science and Computational Intelligence (IJSSCI), IGI-Global, 7 (2), 2015.
7) J. Sarivougioukas, A. Vagelatos: Use of Denotational Mathematics for the Formal Description of Autonomous Migration and Polymorphism as Prerequisites for Mobility in Home UbiHealth, 14th IEEE International Conference on Cognitive Informatics & Cognitive Computing, July 2015, Beijing, China.
8) J. Sarivougioukas, A. Vagelatos: The necessity for ubiquitous process-awareness in the contemporary Hospital. Invited speech, 17o Pan-Hellenic Conference in Healthcare Management, Oct 16-17, 2015, Athens, Greece.
9) J. Sarivougioukas, A. Vagelatos and I. Lagaris: Prerequisites of autonomous software operations in Home UbiHealth described formally with Denotational Mathematics, International Journal of Software Science and Computational Intelligence (IJSSCI), IGI-Global, 8 (1), 2016.

AUTHOR’S RESUME

Received a B.Sc. in Electrical Engineering, 1984, Ohio University, Ohio, USA, then in 1995 obtained a M.Sc. in Computer Engineering, Manhattan College, New York, USA, and in 1997 obtained a M.Sc. in Electrical Engineering from Manhattan College, New York, USA. Taught as an Adjunct Professor at the Computer Science Department of the City University of New York, City College, at the Information Systems Department of the Borough of Manhattan Community College, and at the Electrical Engineering Department of Manhattan College, New York, USA. Designed and implemented IT Systems in the area of Healthcare such as the National Healthcare Network (ESYnet) and the National Disease Related Group (DRG) reimbursement healthcare system for the National Healthcare System of Greece. Worked at large hospitals as computer and electrical engineer for more than 20 years. Applied international standards for the certification of Quality Systems, the design and licensing of medical devices, and the evaluation of biomedical technology. The research interests include ubiquitous computing, Systems Theory, and Cybernetics in Healthcare and Social Security.