UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2569

A REFERENCE ARCHITECTURE FOR HEALTHCARE SUPPORTIVE HOME SYSTEMS: MISSIONS ESTABLISHMENT AND VALIDATION

LINA MARIA GARCÉS RODRÍGUEZ
FLAVIO OQUENDO
ELISA YUMI NAKAGAWA

Nº 417

RELATÓRIOS TÉCNICOS

São Carlos – SP

Fev./2017

A Reference Architecture for Healthcare Supportive Home Systems: Missions Establishment and Validation

Lina Maria Garcés Rodríguez
E-mail: linamgr@icmc.usp.br
Department of Computer Systems
University of São Paulo - USP
São Carlos, Brazil

IRISA Research Institute
University of South Brittany - IRISA/UBS
Vannes, France

Prof. Dr. Flavio Oquendo
E-mail: flavio.oquendo@irisa.fr
IRISA Research Institute
University of South Brittany - IRISA/UBS
Vannes, France

Prof. Dr. Elisa Yumi Nakagawa
E-mail: elisa@icmc.usp.br
Department of Computer Systems
University of São Paulo - USP
São Carlos, Brazil

ICMC-USP
São Carlos

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Ambient Assisted Living (AAL) domain aims to address the difficulties that the demographic shift is originating, creating technologies to support activities of daily life (ADL) of elders. Specifically, Healthcare Supportive Home (HSH) systems have been proposed to overcome the increasing demand for telehomecare, due to the growing of diseases and disabilities, and to avoid long-term hospitalization. HSH systems involve a variety of technologies and aim to provide an autonomous life in their residence to patients suffering of chronic disease, handicaped persons, and dependent elderly people [14]. HSH systems are considered as Systems-of-Systems (SoSs).

An SoS is a new type of system conformed by independent systems, e.g., in the context of HSH systems they are constituted by monitoring, telemedicine, rehabilitation, domotic and electronic health records systems. SoSs have raised in response to needs of achieving complex missions that can not be addressed by individual systems. Hence, complex missions are fulfilled by the SoS behaviors that emerge from the interaction among their constituent systems. The early establishment and validation of SoSs missions can support the better understanding of their emergent behaviors and the definition of expected capabilities from constituent systems.

In another perspective, reference architectures (RAs) are basis for the development, evolution, and standardization of software systems in specific domains, and can contain missions and emergent behaviors of these systems. In this context, RAst can orient the construction of SoS in an efficient way.

In this technical report we presented results of establishing and validating missions of HSH Systems in a Reference Architecture. For this, we use a systematic process [**?**] to establish, specify and validate SoSs missions contained in RAs. As result, the RA can provide complete and congruent models representing HSH systems missions, emergent behaviors, and objects (i.e., entities and events) of the SoSs in the domain, as well as constituent systems responsibilities and their communicational and operational capabilities. We intend such RA can better support the development, evolution, and standardization of HSH systems.

The remainder of this report is organized as follows. In Section 2 we present the process to establish and validate SoSs missions in RAs. Section 3 describes results of applying such process in a RA for HSH systems. Finally, Section 4 details conclusions and future works.

Chapter 2

Process to Establish and Validate SoSs Missions in RAs

In this section we present the process to define missions of SoSs at RA level. Our process was inspired by several initiatives from areas of goal-oriented requirements engineering (GORE) [1, 20, 11], missions-oriented SoS designing [6], and reference architecture engineering [8]. The process consists of ten phases as depicted in Figure 2.1. Information and models flows are represented as dashed arrows, while the process flow is symbolized as bold arrows.

During the missions establishment, several domain information are generated. Moreover, mKAOS models to represent missions, emergent behaviors, objects, and constituents systems capabilities are produced. We used mKAOS approach since it is the only mission oriented tool, of we are aware, to model domain knowledge in the SoS context. It is important to highlight that this process is oriented to the first three types of SoSs, i.e., directed, acknowledged and collaborative, whose missions can be defined. The process is not recommended to establish missions in RAs for virtual SoSs, since they lack of a common purpose [21]; hence, it is difficult to establish missions for those SoSs, and their constituent systems and their capabilities are difficult to discern and distinguish [21]. The remainder of this section details process phases.

2.1 Phase 1 - Identification of information sources

As defined by ProSA-RA [8], the first step to create any RA is the definition of the target domain. After, the principal sources of domain knowledge are identified. The objective of

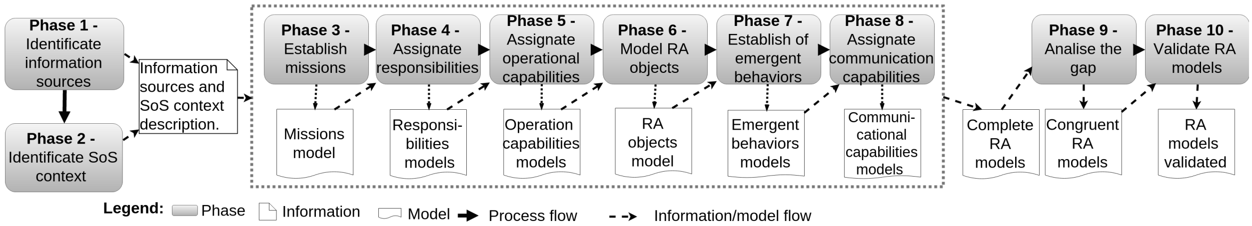


Figure 2.1: Process to establish missions for RAs of Sos.

this phase is to obtain the basis to guide the definition of missions for SoSs in the specified domain. Several information sources can be considered [8]:

- <u>Stakeholders:</u> All people who have some interest or role in the SoS, e.g., final users, customers, project managers, software architects, developers, or partner organizations. They could guide the selection of SoSs missions contained in the RA.

- <u>Domain guidelines:</u> Documents describing processes, activities, or tasks in the domain could assist the identification of SoSs emergent behaviors and capabilities of constituent systems to be considered in the RA.

- <u>Ontologies, taxonomies, reference models:</u> The use of formal domain terminology facilitates the further understanding of missions in RAs.

- <u>Quality models:</u> Well defined quality attributes for software systems in the domain, or in related domains, could orient the selection of important non-functional properties of SoSs that need to be contemplated in the RA.

- <u>Concrete SoSs in the domain:</u> Important SoSs in the domain can be analysed to identify common global missions and possible emergent behaviors. Moreover, common constituent systems, their interactions, responsibilities, and capabilities could be also determined. Such analysis can be performed using SoSs documentation and during the SoSs operation, if possible.

- <u>Reference architectures:</u> Documentation of other RAs, established for the same or related domains, could be used as a knowledge repository to identify the aforementioned information sources.

Information sources can be identified through expert suggestions, previous knowledge, or systematic literature reviews.

2.2 Phase 2 - Identification of the SoSs context in the domain

This phase aims to define the SoSs type that must be considered by the RA. Moreover, missions of concrete SoSs and the generic constituent systems are also characterized.

2.2.1 Definition of SoSs type

The early definition of the SoSs type (i.e., directed, collaborative or acknowledged) in the RA is important, since it guides the selection of architecting principles [3]. The SoSs type determines the level of adaptation and cooperation of constituent systems with respect to individual missions, capabilities, and entities. This activity can be supported by concrete SoSs documentation and domain literature.

2.2.2 Identification of SoSs missions in the domain

Considering that most of concrete SoSs specifications are not mission-oriented, reengineering of missions from requirements is needed. If concrete SoSs are mission oriented, this activity is not necessary. Hence, requirements of a set of concrete SoSs are extracted from their documentation. Goals for each SoS are established searching for intentional keywords, such as “objective”, “purpose”, “intent” and “aim” in SoSs documentation [11]. Based on defined goals for each SoS, missions are established following the mKAOS approach [6]. Here, it is important to characterize which missions are of the SoS and which ones of their constituent systems. Moreover, common individual missions of constituent systems that are part of the collection of SoSs are abstracted. This abstraction must be independent of the platform, technology, organization, and other specific details. For each abstraction, generic constituent systems are created. Documentation of concrete SoSs in the domain, and possibly of RAs in related domains, can be analysed to identify the SoSs types, their missions and their generic constituent systems.

2.3 Phase 3 - Establishment of SoSs missions in the RA

The objective of this phase is to determine all possible missions for any SoS in the domain. Two approaches, i.e., refinement and abstraction, are used to define the SoSs missions in the RA. Refinement allows to establish missions starting from high-level SoSs missions, until reach constituent systems individual missions. Whilst, missions abstraction start from individual missions to global SoSs missions. Both approaches are used iteratively to provide a comprehensive and congruent set of missions in the RA.

2.3.1 Refinement of missions

The generic pattern showed in Figure 2.2 can be used as a start point to define the SoSs missions in the RA. This pattern establishes that any SoS in the domain has the global mission (GM) of satisfying the stakeholder needs. To accomplish GM, the SoS must address both missions of satisfying functional (GM1) and non-functional (GM2) needs.

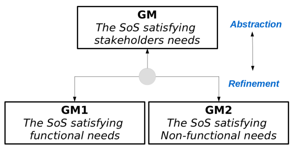


Figure 2.2: General missions pattern. Adapted from [20].

GM1 and GM2 must be refined in more concrete missions. Concrete missions are identified by asking **HOW** the high-level missions can be reached, e.g., GM is addressed through the achievement of GM1 and GM2. Refinement must be performed until reach individual missions, i.e., fine-grained missions that can be assigned to generic constituent systems. Refinement patterns [25] proposed by the GORE area can be used to orient this activity. Refinement of SoSs missions is supported by missions of concrete SoSs identified in Phase 2. However, it is possible that vocabulary inconsistencies exist between concrete SoSs missions; hence, domain ontologies, taxonomies or reference models can be used to establish a common terminology for the RA [8]. Moreover, domain guidelines detailing process or activities can orient the identification of new missions that were not considered in concrete SoSs. Similarly, domain quality models can aid the refinement of GM2, since they define taxonomies of quality attributes for systems in the domain.

2.3.2 Abstracting new missions

Missions identified in Phase 2 can be abstracted by asking **WHY** questions, i.e., why is this mission important? Concrete missions are justified by at least a high-level mission, this is, a concrete mission must contribute to achieve a more general SoSs mission in the domain. For example, in Figure 2.2, GM1 and GM2 are justified because together address GM. Domain ontologies, guidelines and quality models can support this activity.

2.3.3 Modeling SoSs missions in the RA

The missions model for the RA is made. mKAOS mission models [6] can be used to represent the missions in the RA. In those models, missions are structured as a tree where leaf nodes represent individual missions of generic constituent systems, non-leaf nodes represent SoSs missions, and roots missions represent global missions, i.e., GM, GM1 and GM2. Refinement, abstraction and modelling of missions are parallel and iterative activities that stop when the next two criteria are achieved:

- **Completeness criterion 1:** The RA mission model is complete with respect to the abstraction relationship, if only if, each mission, with exception of individual missions, is refined at least by two concrete missions.

- **Completeness criterion 2:** The RA mission model is complete with respect to the refinement relationship, if only if, each individual mission can be assigned to a generic constituent system.

2.4 Phase 4 - Assignation of responsibilities to generic constituent systems

In this phase, individual missions, i.e., leaf nodes missions in the RA missions model, are assigned to generic constituent systems. An individual mission must be of responsibility of only one generic constituent system. However, one generic constituent system can have several individual missions under its responsibility. If any of the generic constituents systems identified in Phase 2 can address such individual mission, to create a new generic constituent system responsible of such mission. This phase is considered complete if the next criterion is achieved:

- **Completeness criterion 3:** The RA responsibility model is complete if and only if, all individual missions are placed under the responsibility of one and only one generic constituent system.

Descriptions of all responsibilities of each generic constituent system can be made using the responsibility models of mKAOS [6].

2.5 Phase 5 - Assignation of operational capabilities to generic constituent systems

In this phase the operational capabilities of each generic constituent system are defined. An operational capability is a responsibility of the generic constituent system. Such capabilities can be well known or uncertain, depending if the SoSs considered in the RA are directed, collaborative or acknowledged. Hence, the specification of operational capabilities is restricted by the operational independence level of the generic constituent systems participating of the SoSs. For instance, if the RA addresses directed SoSs, operational capabilities of their generic constituent systems are limited to the responsibilities assigned in Phase 4. If the RA is oriented to acknowledged or collaborative SoSs, generic constituent systems can have more operational capabilities than the expected responsibilities; hence, for such RA, all operational capabilities can not be determined.

For each generic constituent system it is created an operation capability model, detailing all operations that are under their responsibility, and that are expected to be executed to achieve their individual missions, and hence, the SoSs missions. mKAOS operational models [6] can be used to support this activity. This phase finished when the following criterion is achieved:

- **Completeness criterion 4:** The RA operational capability models are complete if, for each generic constituent systems, each expected responsibility is modeled as an operational capability.

However, it does not mean that extra operational capabilities can not be included in the model.

2.6 Phase 6 - Modeling SoSs objects in the RA

This phase aims to define operational SoSs objects, i.e., entities or events, in the RA. Entities implement a data structure that represents some element of the physical world, and events are related to specific circumstances to which SoSs must react [6]. For each operational capability of each generic constituent system, entities and events are identified. One instance of each entity and event must be selected, since several operational capabilities can use the same entity or event. Object models can be supported by mKAOS.

2.7 Phase 7 - Establishment of SoSs emergent behaviors in the RA

SoSs missions are fulfilled by emergent behaviors that are consequence of interactions between SoSs constituent systems; hence, such behaviors can not be designated as a capability of individual constituent systems. The knowledge about an emergent behavior is spread across the entire SoS and exists only at the macro-level [28]. Moreover, it is possible that new knowledge exists at a higher level of abstraction itself and there are not enough means to break this new knowledge to be attributed to any constituent system's behavior [28]. In this phase, SoSs emergent behaviors in the RA are identified and modeled.

2.7.1 Assignation of emergent behaviors to missions

The main idea is to designate an emergent behavior by each high-level mission defined in the RA mission model; hence, all expected missions could be executed. The assignation can be made following a bottom-up approach, starting to define an emergent behavior to the parent missions of individual missions, until the general missions GM1, GM2, and finally GM. As result, a tree of emergent behaviors of the RA is defined.

2.7.2 Identification of SoSs objects involved in emergent behaviors

For each emergent behavior, to define their input and output entities/events, following a bottom-up approach. Objects that were established in Phase 6 and that are generated by generic constituent systems are the input objects for low-level emergent behaviors. Specifically, for a low-level emergent behavior, which is directly related with a specific low-level mission, to select the objects generated by the generic constituent systems involved in such misson as input objects of such behavior. After, to identify possible output objects, which can be aggregated objects for the low-level emergent behavior. For high-level behaviors, input objects are the output objects of low-level behaviors. Hence, more abstract objects are identified when the abstraction level of emergent behaviors increases. In this perspective, knowledge that is not known by the generic constituent systems is generated in order to fulfill high-level missions.

The responsibilities, operational capabilities and object models can support this activity. Moreover, domain guidelines and documentation of dynamic characteristics of concrete SoSs can be used to identify the objects involved in emergent behaviors.

2.7.3 Modeling emergent behaviors

Objects participating of an emergent behavior must be communicated by constituent systems or generated through the execution of other emergent behaviors. For each input/output entity involved in an emergent behavior it is necessary to define a communicational capability. RA models of emergent behaviors are made, containing the communicational capabilities that are needed to execute such behaviors and, also, that are result of their execution. mKAOS emergent behaviors model can support this activity.

2.7.4 Updating RA models

High-level entities or events are identified as result of establishing the emergent behaviors, therefore, the RA objects model must be updated. Moreover, new objects of low-level also can be characterized, since it is possible that not all objects related with operational capabilities of generic consituent systems were initially identified. In this case, operational capabilities and responsibility models must be updated.

2.8 Phase 8 - Assignation of communicational capabilities

Output entities of a generic constituent system can be inputs for other systems. In this phase, such interactions are identified and modeled as communicational capabilities of such systems.

2.8.1 Identification of interactions between constituent systems

To execute emergent behaviors interactions generic constituent systems need to interact. Such interactions are considered as communicational capabilities for those systems. In this context, for each interaction between generic constituent systems, to associate a communicational capability for both systems, i.e., one to send the entity and other to receive it, if it was not identified before. Since this interactions represent a process or a workflow that need to be supported by the RA, domain guidelines and concrete SoSs documentation are important information source for their identification.

2.8.2 Modeling the RA communicational capabilities

Interactions between generic constituent systems are modeled using the mKAOS communicational capability model. This model allows to understand the entity communicated, the generic constituent systems who generate it and who use it, and the communication flow.

2.8.3 Updating RA models

Objects, responsibilities, communicational and operational capabilities models can need updates due to new entities possibly identified in this phase.

2.9 Phase 9 - Analysing the gap

RA models must be congruent between them, therefore, in this phase possible gaps between models need to be solved. Hence, the following mappings are performed:

- *Emergent behavior - SoS missions models:* To check if for each high-level mission, it exists an emergent behavior that allows their execution;
- *Communicational capabilities - Emergent behaviors models:* To check if all required communicational capabilities to trigger all emergent behaviors were considered;
- *Generic constituent system - Communicational capabilities models:* To check if all communicational capabilities have just one constituent system responsible for it;
- *Entity/event - Operational/communicational capabilities models:* To check if all entities/events involved in the RA are participating at least of one operational/communicational capability;
- *Entities/events - Objects model:* To check if all entities/events involved in the RA were defined in the object model;
- *Individual mission - Operational capability models:* To check if all individual missions are from responsibility of one constituent system.

As result of this phase, all RA models are complete and congruent and can be validated.

2.10 Phase 10 - Validation of SoSs missions in the RA

The last phase of our process aims to identify if RA models can support the definition of missions, emergent behaviors, objects, and constituent systems capabilities of a concrete SoS in the RA domain.

RA models can be validated through surveying stakeholders. Sets of RA models can be selected to be reviewed by stakeholders according with their interests and expertise. For instance, final users and domain experts can be more comfortable in reviewing missions and responsibilities models, whilst systems engineers, software architects and developers can review objects, capabilities and emergent behaviors models.

Similarly, scenario-based techniques have been used for validating requirements specifications [22] and seems to be a good alternative to assess the reliability of the RA models. Real scenarios of a concrete SoS can be proposed to validate such models. The idea is to instantiate the RA models to represent a concrete SoS that addresses such scenarios. The following bottom-up approach can be used for each scenario: (i) to select a set of constituent systems to conform the concrete SoS based on the generic constituent systems defined in the RA; (ii) to designate real input and output objects for the operational capabilities; (iii) to identify the required capabilities to communicate such objects; (iv) to determine which emergent behaviors defined in the RA can be executed based on the communicational capabilities; and (v) to determine if such emergent behaviors allow to achieve the expected missions for the concrete SoS.

Moreover, this phase allows to identify gaps or inconsistencies that avoid to achieve the SoSs missions specified in the RA. Moreover, possible undesired behaviors can be detected and solved.

Chapter 3

RA for HSH systems: Missions Establishment and Validation

In this section we present results of following the process to establish and validate SoSs missions in a RA for Healthcare Supportive Home (HSH) systems.

3.1 Phase 1 - Identification of information sources

Definition of the target domain The Ambient Assisted Living (AAL) domain aims to address the difficulties that the demographic shift is originating, creating technologies to support activities of daily life (ADL) of elders. Specifically, HSH systems have been proposed to overcome the increasing demand for telehomecare, due to the growing of diseases and disabilities, and to avoid long-term hospitalization. HSH systems involve a variety of technologies and aim to provide an autonomous life in their residence to patients suffering of chronic disease, handicaped persons, and dependent elderly people [14].

Information sources identified Four categories of stakeholders were defined by Hutch et al. [19]: (i) Primary Stakeholders, which are private users of HSH systems, e.g., senior and impaired citizens, or private caregivers; (ii) Secondary Stakeholders, which are professional users of HSH systems, e.g., medical staff; (iii) Tertiary Stakeholders, which are suppliers of HSH systems, e.g., research organisations, enterprises with a business in tele-medicine or tele-care or providers of the IT infrastructure; and (iv) Quaternary Stakeholders:, which are supporters of HSH systems, e.g., policy-makers or social (and private) insurance companies.

International guidelines for chronic conditions management were considered [23] offering processes to treat patients with chronic conditions. Such guidelines were suggested by a clinical nurse specialist in chronic diseases. Moreover, we used the K4CARE (Knowledge-Based HomeCare eServices for an Ageing Europe) ontology [27]. Such ontology includes hierarchies of concepts related to the syndromes of cognitive impairment and immobility of homecare patients.

We searched for concrete HSH systems in the AAL Joint Programm[1] and Cordis[2] projects bases. We identified three projects, named GiraffPlus[3], CHRONIOUS[4] and Dem@care[5] proposing alternatives of HSH systems.

Two reference architectures proposed for the AAL domain, specifically for healthcare, were considered: Feelgood [12] and Continua [10]. Both RAs were identified through conducting a systematic literature review [15]. Finally, we used the QM4AAL, a quality model established for the AAL domain [16], which defines requirements of quality attributes for software systems in such domain.

3.2 Phase 2 - Identification of the SoSs context in the domain

Definition of the SoS type HSH systems are acknowledged SoSs, since the constituent systems maintain their independent management and operation. Examples of constituent systems are smart homes, domotic systems, electronic health records (EHR) systems, monitoring systems, smart devices such as smart Tvs or set-top-boxes, and rehabilitation systems. Constituent systems collaborate with the HSH system to achieve and evolve the global mission.

Identification of missions of the HSH systems We defined the missions and the responsibility models for the three concrete HSH systems, CHRONIOUS, GiraffPlus and Dem@care, which were identified in Phase 1. Figure 3.1 shows an excerpt of both models for the CHRONIOUS system.

Moreover, based on common capabilities offered by constituent systems in such concrete HSH systems, we identified several generic constituent systems, such as, physical parameters monitors, rehabilitation support systems, EHR systems, and emergency systems.

[1]http://www.aal-europe.eu/
[2]http://cordis.europa.eu/home_en.html
[3]http://www.giraffplus.eu/
[4]http://www.chronious.eu/
[5]http://www.demcare.eu/

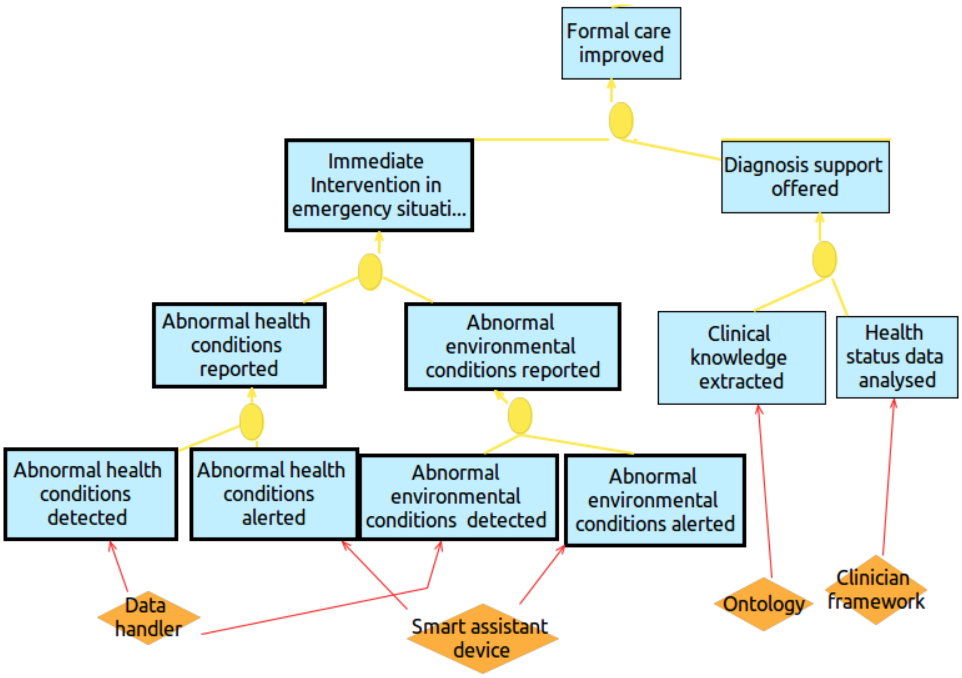


Figure 3.1: Excerpt of the mission and responsibility models for the CHRONIOUS system.

3.3 Phase 3 - Establishment of SoSs missions in the RA

We used the general mission pattern illustrated in Figure 2.2. Based on such pattern we refine the missions in the RA. Missions models of concrete HSH systems, guidelines to manage chronic diseases [23], RAs for healthcare systems [15], the K4CARE ontology [27] and the QM4AAL quality model [16] gave the foundations needed to establish the missions tree for HSH systems in the RA. Figure 3.2 depicts a fragment of the mission model, showing the missions of more high-level of the RA for HSH systems.

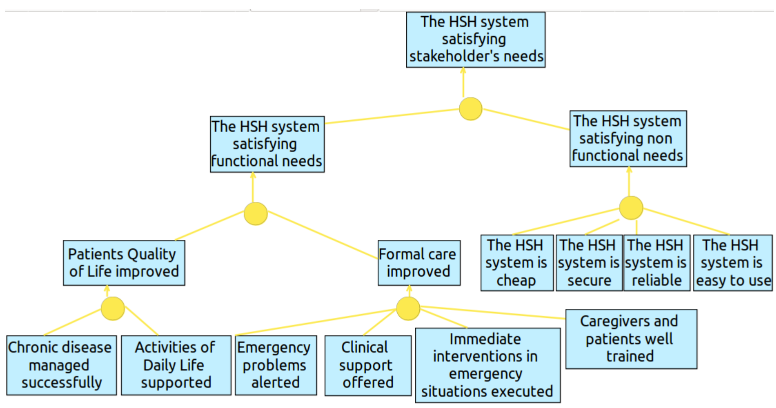


Figure 3.2: High-level missions model of the RA for HSH systems.

To achieve the stakeholders functional needs, the patients Quality of Life (QoL) and the formal care must be improved. Patients QoL is improved through a successful management of chronic diseases and offering support to patients Activities of Daily Life (ADL). Formal care is improved when emergency problems are alerted, clinical support is offered, immediate interventions in emergency situations are executed, and caregivers and patients are trained. To satisfy stakeholders non-functional needs, the HSH systems must be cheap, secure, reliable and easy to use.

We applied refinement and abstraction methods until achieve the two completeness criteria described in Section 2.3.3. Hence, all high-level missions have at least two sub-missions and low-level missions are the individual missions assigned to generic constituent systems.

Figures 3.3, 3.4,3.5 and 3.6 show refinements of the RA missions models. In total 283 missions were established in the RA, corresponding to 37 missions for HSH systems and 246 individual missions for their generic constituent systems.

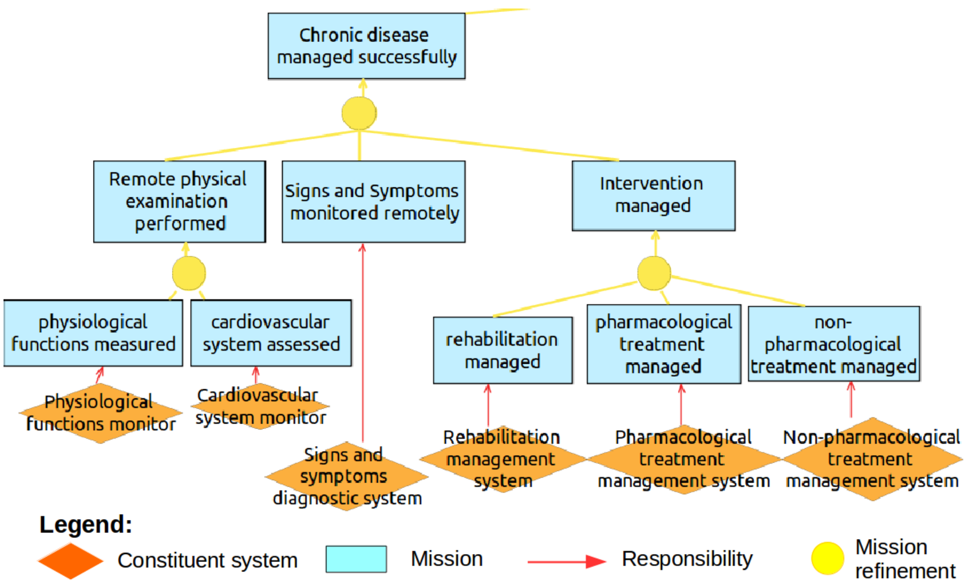


Figure 3.3: Excerpt of the mission and responsibility models of the RA for HSH systems. Refinement of the *Chronic disease successfully managed* high-level mission.

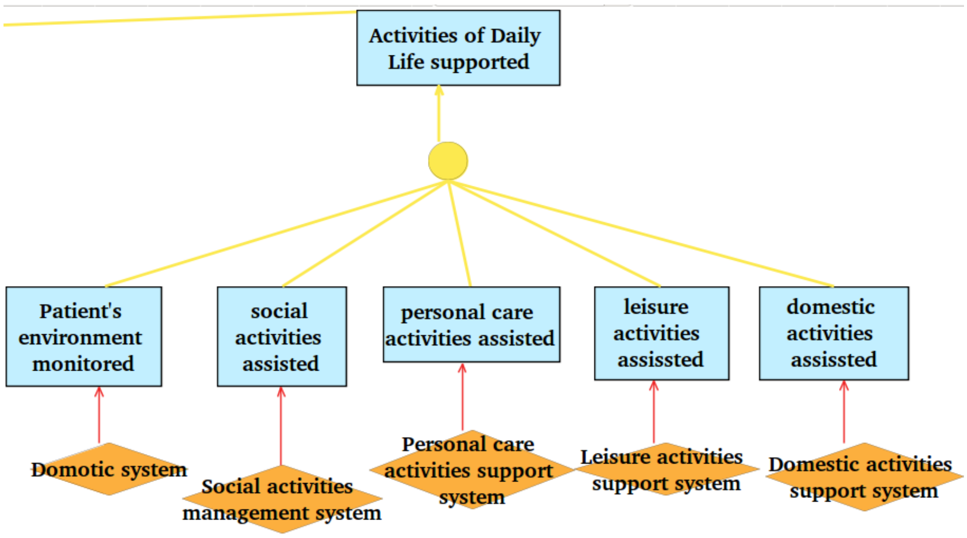


Figure 3.4: Excerpt of the mission and responsibility models of the RA for HSH systems. Refinement of the *Activities of Daily Life Supported* high-level mission.

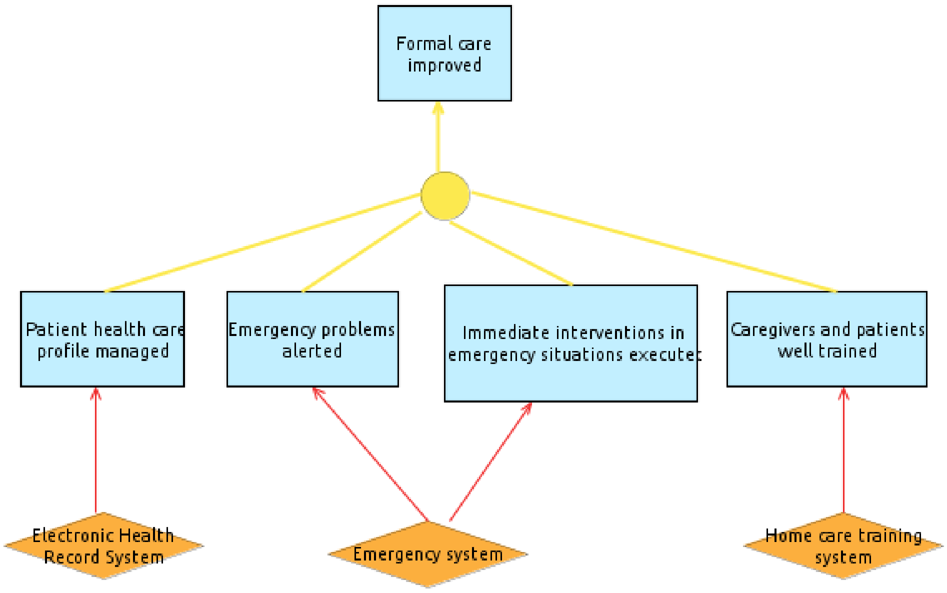


Figure 3.5: Excerpt of the mission and responsibility models of the RA for HSH systems. Refinement of the *Formal Care Improved* high-level mission.

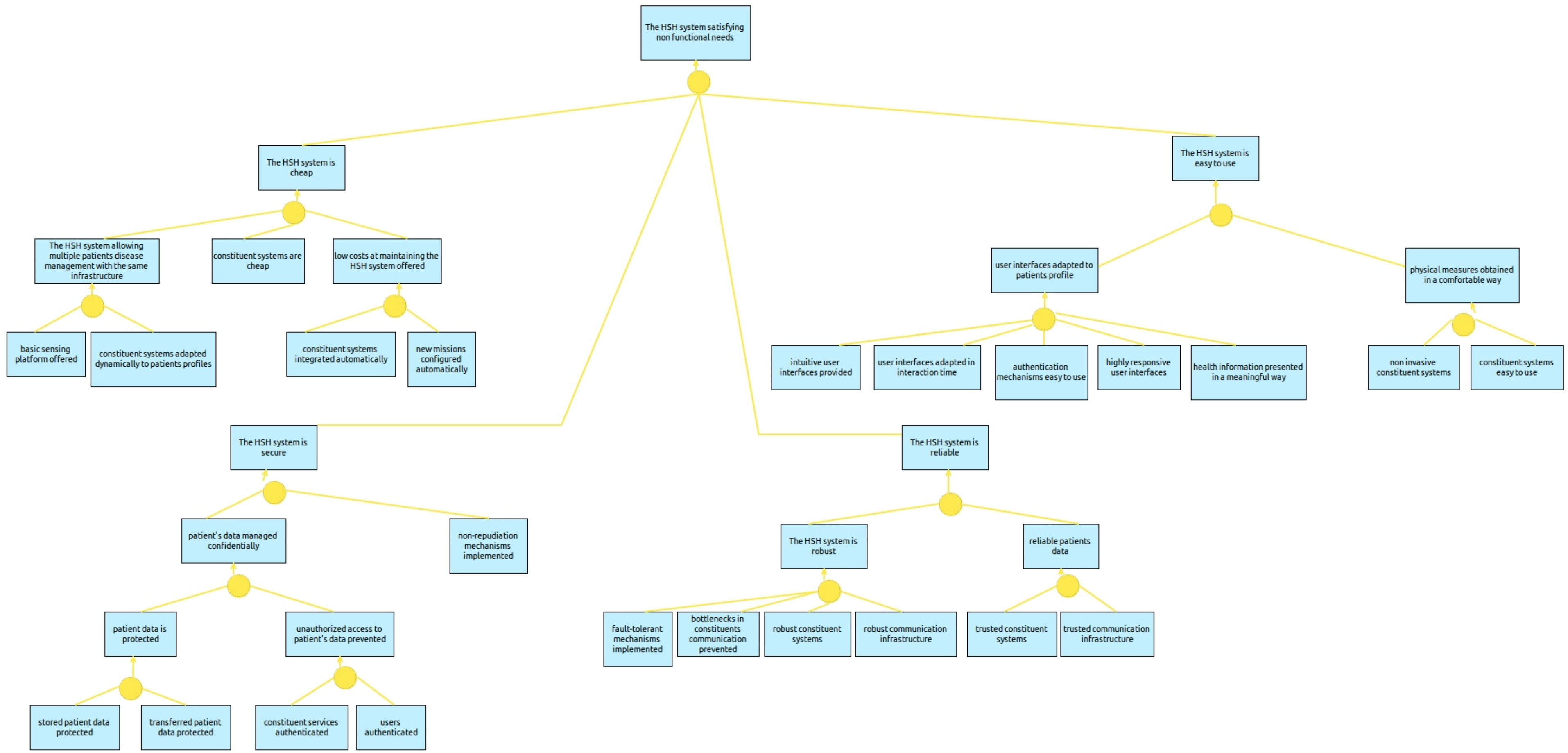


Figure 3.6: Excerpt of the missionmodel of the RA for HSH systems. Refinement of the *HSH Satisfying Non Functional Needs* high-level mission.

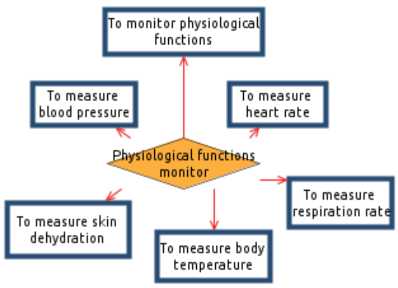
3.4 Phase 4 - Assignation of responsibilities to generic constituent systems

Individual missions were assigned to the generic constituent systems identified in Phase 2. We applied the completeness criterion 3, which was introduced in Section 2.4. Additional generic constituent systems were defined, e.g, sign and symptoms diagnostic systems and pharmacological and non-pharmacological treatment management systems, were identified based on RAs for healthcare and guidelines to manage chronic diseases. Moreover, refinement of generic consituent systems was made to assignate individual missions, e.g., the generic constituent "physical parameters monitors", which was defined in Phase 2, was refined into "cardiovascular system monitor", "physiological functions monitor" as presented in Figure 3.3 and that are related with the *Remote physical examination performed* mission.

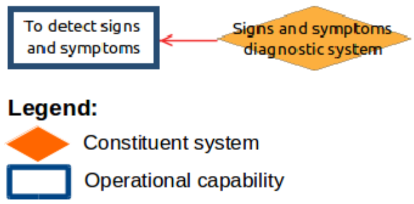
3.5 Phase 5 - Assignation of operational capabilities to generic constituent systems

For each generic constituent system with assigned responsibilities, the expected operational capabilities to achieve the individual missions were assigned and modeled, following the completeness criterion 4 presented in Section 2.5. Moreover, additional operational capabilities were identified based on processes detailed in the guidelines for managing chronic diseases and in the K4CARE ontology.

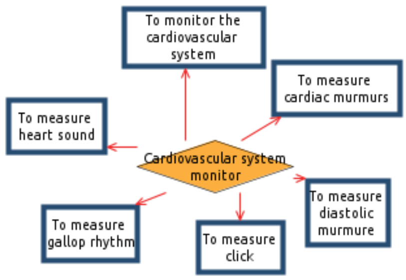
Figures 3.7a, 3.7b and 3.7c depicts the operational capabilities model for the generic constituent systems of: physiological functions monitor, signs and symptoms diagnostic system and cardiovascular system monitor respectively. For instance, to achieve the individual mission *cardiovascular system assessed*, the generic constituent system *cardiovascular system monitor* must perform the operational capability *to monitor the cardiovascular system*. Additional capabilities can be used by the HSH systems to improve their missions achievement. Complete constituent systems operational capabilities models of the RA for HSH systems are presented in Figures 3.8 and 3.9.



(a) Operational capabilities for the generic constituent system: physiological functions monitor.



(b) Operational capabilities for the generic constituent system: signs and symptoms diagnostic system.



(c) Operational capabilities for the generic constituent system: cardiovascular system monitor.

Figure 3.7: Operational capabilities models for generic constituent systems.

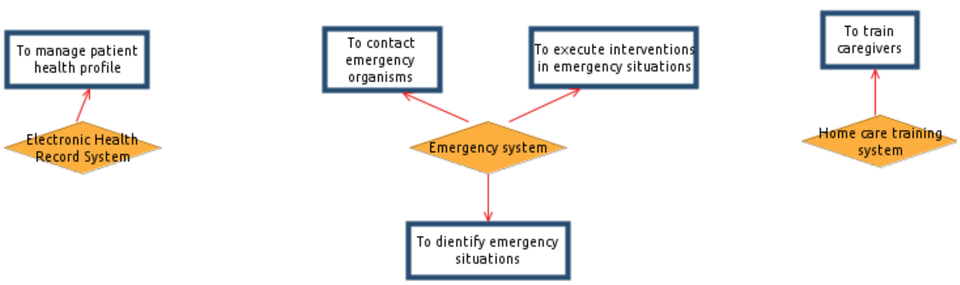


Figure 3.8: Operational capabilities of constituent systems. Second Part.

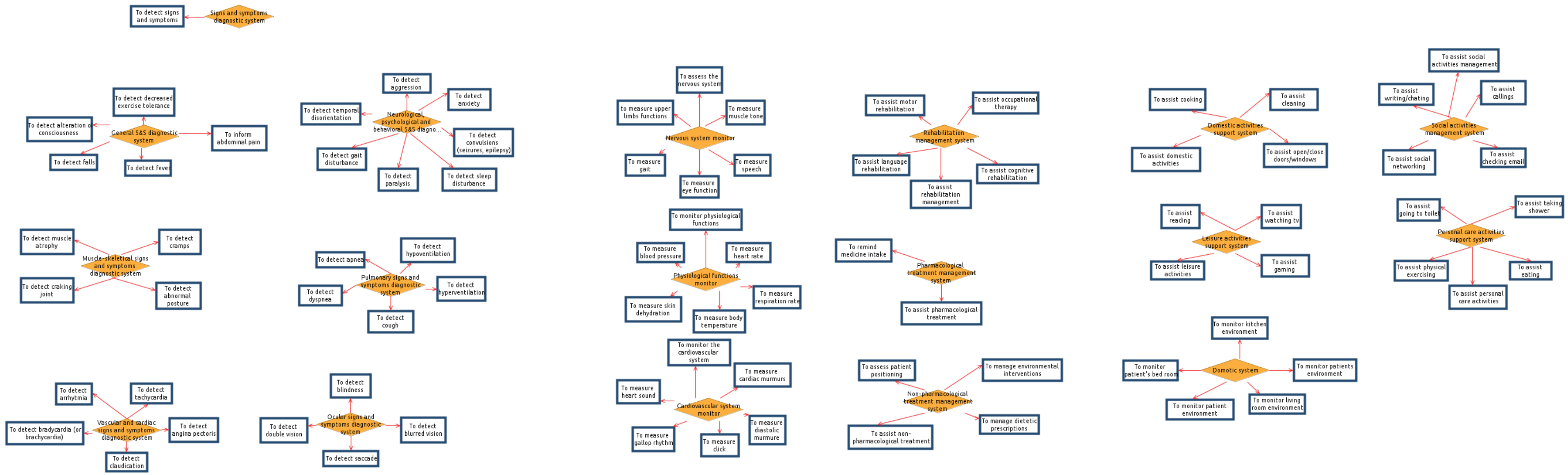


Figure 3.9: Operational capabilities of constituent systems. First Part.

3.6 Phase 6 - Modelign SoSs objects in the RA

For each operational capability defined for the generic constituent systems, we identified the operational objects, i.e., entities and events that are delivered and used by constituents systems of the HSH systems. Figure 3.10 depicts the RA objects model. White boxes represent entities generated by the generic constituent systems and gray boxes are the events executed by such systems. For instance, Table 3.1 presents some input/output entities involved in the operational capabilities of the three constituent systems showed in Figure 3.7.

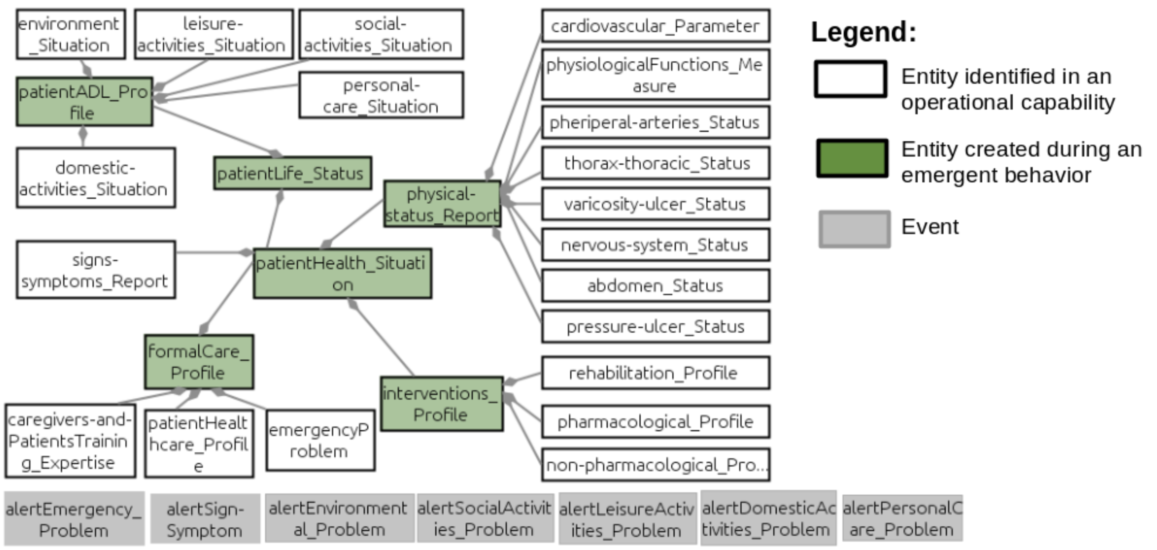


Figure 3.10: Objects model of the RA for HSH systems.

Table 3.1: Examples of input/output entities of operational capabilities.

Constituent system	Operational capability	Input entity	Output entity
Cardiovascular system monitor	To monitor the cardiovascular system		cardiovascular _Parameters
Physiological functions monitor	To monitor physiological functions		physiologicalFunctions_Measures
Signs and symptoms diagnostic system	To detect signs and symptoms	physical-status_Report	signs-symptoms_Report

3.7 Phase 7 - Establishment of SoSs emergent behaviors in the RA

We established an emergent behavior for each mission and detailed the communicational capabilities needed to allow the emergent behavior (i.e., input communicational capabilities) and those capabilities resulting of executing such behaviors (i.e., output communicational capabilities). Figure 3.11 describes some of the emergent behaviors specified, the related missions, and the input/output communicational capabilities involved in such behaviors.

High-level mission	Emergent behavior	Input objects	Communicational capability input	Output objects	Emergent Communicational capability output
Remote physical examination performed	To establish patient's physical status	physiologicalFunction_Measure cardiovascular_Parameter nervous-system_Status thorax-thoracic_Status pheriperal-arteries_Status abdomen_Status varicosity-ulcer_Status pressure-ulcer_Status	To provide physiological functions measures To provide cardiovascular parameters To provide nervous system status To provide thorax and thoracic status To provide pheriperal arteries status To provide abdomen status To provide varicosity ulcer status To provide pressure ulcer status	patientPhysical_Status	To provide patient physical status
Intervention managed	To establish interventions status	rehabilitation_Profile pharmacological_Profile non-pharmacological_Profile	To provide rehabilitation profile To provide pharmacological profile To provide non pharmacological profile	interventions_Profile	To provide interventions profile
Chronic disease managed successfully	To establish patient's health status	Sign-and-symptom patientPhysical_Status interventions_Profile	To provide sign and symptoms To provide patient physical status To provide interventions profile	patientHealth_Status	To provide patient health situation
Activities of Daily Life supported	To establish patient's ADL status	environment_Situation social-activities_Situation personal-care_Situation leisure-activities_Situation domestic-activities_Situation	To provide environment situation To provide social activities situation To provide personal care situation To provide leisure activities situation To provide domestic activities situation	patientADL_Profile	To provide patient ADL profile
Patient's QoL improved	To establish patient's QoL level	patientHealth_Status patientADL_Profile	To provide patient health situation To provide patient ADL profile	patientLife_Status	To provide patient life status
Formal care improved	To execute formal care	patientHealthcare_Profile alertEmergency_Problem caregivers-and-PatientsTraining_Expertise	To provide patient healthcare profile To alert emergency problem To provide expertise level of caregivers and patients	emergencyInterventions_Plan formalCare_Profile	To provide emergency intervention plan To provide formal care profile

Figure 3.11: Relation between missions and emergent behaviors, and the involved communicational capabilities.

Moreover, Figure 3.12 shows an excerpt of the RA emergent behaviors model that represent the emergent behaviors, missions and communicational capabilities expressed in Table **??**. Emergent behaviors are represented by orange boxes, communicational capabilities of generic constituent systems are symbolized as boxes with blue edges and boxes with black edges represent communicational capabilities that emerge as result of new knowledge generated by the HSH systems. Entities created during the emergent behaviors were included in the object model, and are represented as green boxes in Figure 3.10. Figure 3.13 details the complete emergent behaviors model of the RA for HSH systems.

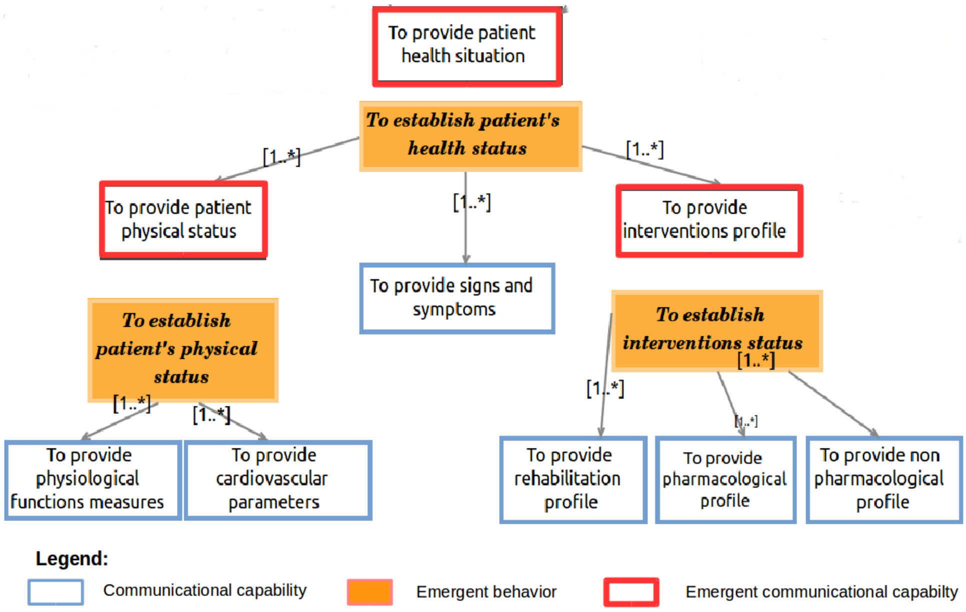


Figure 3.12: Emergent behaviors model excerpt of the RA for HSH systems.

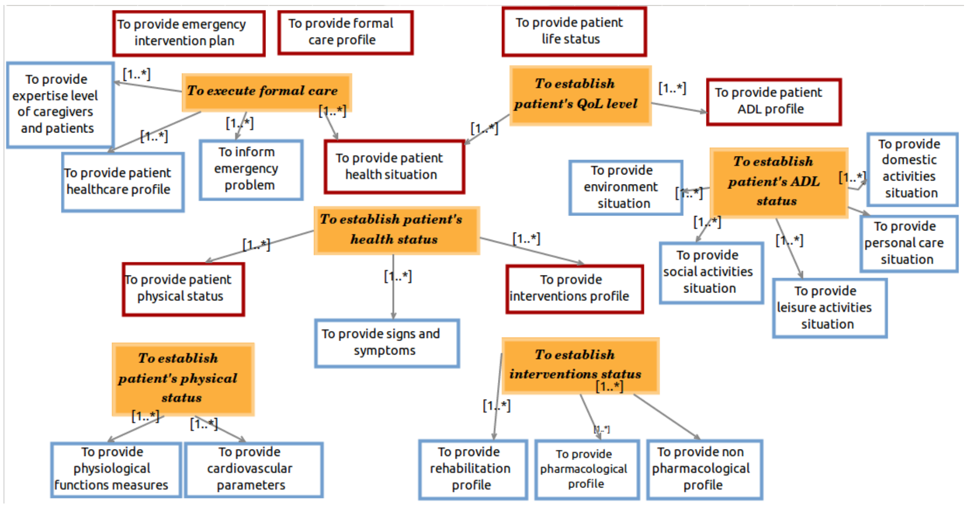


Figure 3.13: Emergent behaviors model of the RA for HSH systems.

3.8 Phase 8 - Assignation of communicational capabilities

Interactions between generic constituent systems were identified. We considered the input entities, and the related communicational capabilities, needed by each operational capability of generic constituent systems. For each interaction, a communication capability model was made, containing involved generic constituent systems, entities exchanged, events generated and communicational capabilities involved. Figure 3.18 presents one communicational capability model of the RA for HSH systems. This model represents the interactions needed to achieve two operational capabilities: "to detect signs and symptoms" and "to manage patient health profile", of the generic constituent systems *signs and symptoms diagnostic system* and *electronic health record system* respectively. Such interaction between both systems are needed, since the entity "signs-symptoms_Report" communicated by the *signs and symptoms diagnostic system* is required by the *electronic health record system* to manage the patient health profile (i.e., an operational capability of such system). Additionally, to manage such profile, the *electronic health record system* needs other two entities: the "physical-status_Report" and the "interventions_Profile" that are generated by two communicational capabilities of the HSH systems (i.e., "to provide patient physical status" and "to provide interventions profile") as result of their emergent behaviors: "to establish patient physical status" and "to establish intervention status", which were introduced in Table **??**. Moreover, the entity "physical-status_Report" is also required by the *signs and symptoms diagnostic system* to create the "signs-symptoms_Report" entity, as defined in Table 3.1.

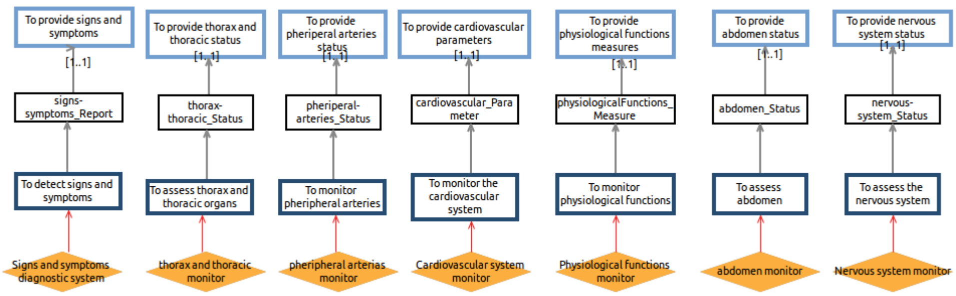


Figure 3.14: Communicational capabilities models of the RA for HSH systems. Part 1

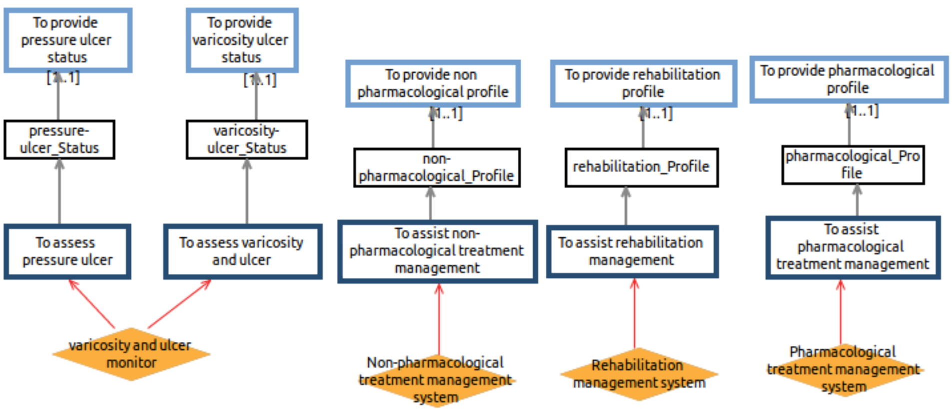


Figure 3.15: Communicational capabilities models of the RA for HSH systems. Part 2

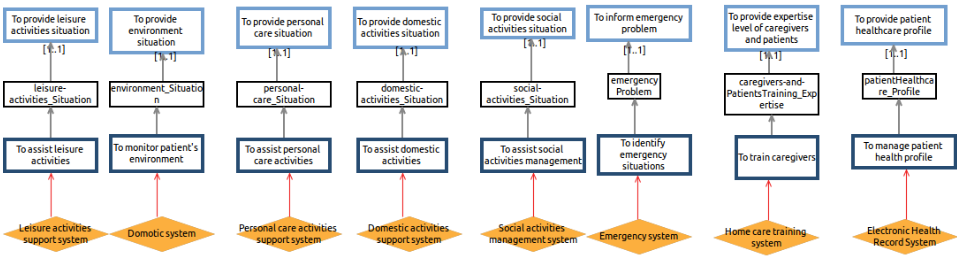


Figure 3.16: Communicational capabilities models of the RA for HSH systems. Part 3

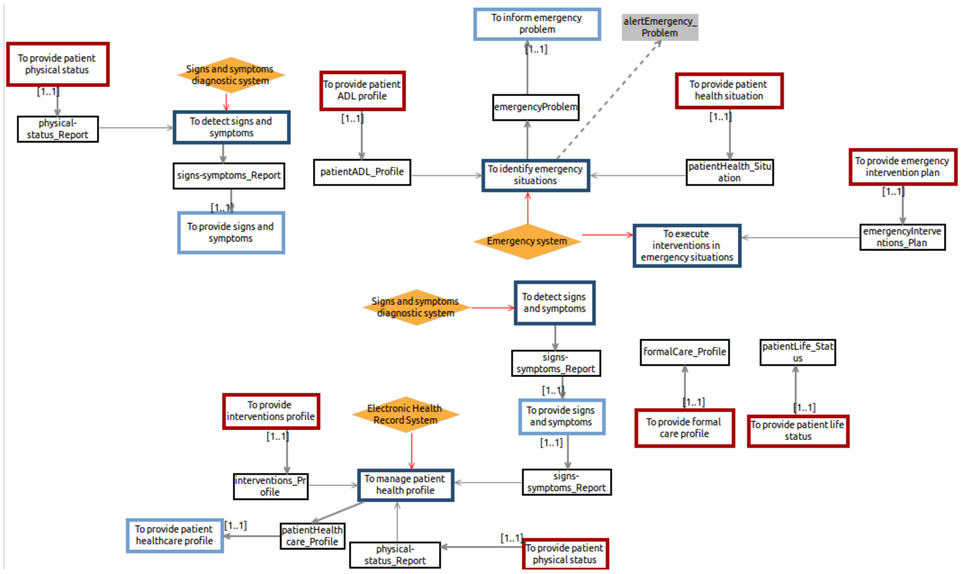


Figure 3.17: Communicational capabilities models of the RA for HSH systems. Part 4.

3.9 Phase 9 - Analysing the gap

We performed the mapping between the RA models and we identified several gaps that were solved without difficulty. As result of this phase, models of the RA for HSH systems are complete and congruent.

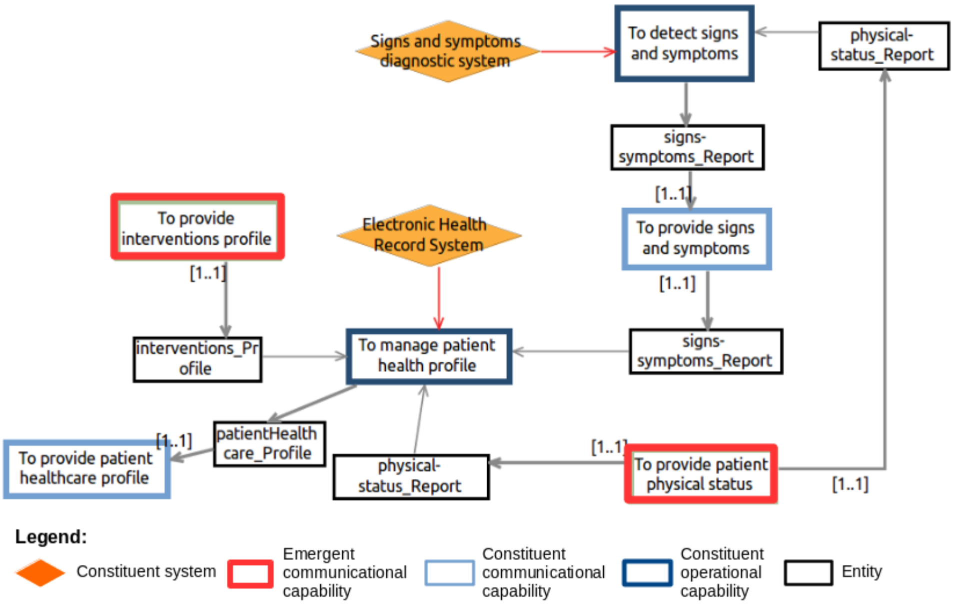


Figure 3.18: Model for the communicational capability: *To provide patient healthcare profile*.

3.10 Phase 10 - Validation of SoSs missions in the RA

To validate the RA models for HSH systems, we selected one scenario of those proposed by the BRAID project (See page 11 in [17]). In short, the scenario describes daily activities of Patricia, a 65 years old lady with sleep problems, diabetes type 2 and bouts of depression. We analysed the scenario and established the missions model for a concrete HSH system that can support Patricia's activities of daily life. Moreover, concrete constituent systems were identified and related with generic constituent systems defined in the RA. Responsibilities, operational and communicational capabilities (and their related objects) of constituent systems were identified based on the RA models. Moreover, emergent behaviors for the concrete HSH system were also defined using the emergent behaviors model of the RA. Models for the concrete HSH system and their mapping with the RA models are presented in Figures 3.19, 3.20, and 3.21. We identified that functionalities reported by the BRAID's scenario can be achieved by the concrete HSH system that was defined through the instantiation of RA models. In this perspective, we can said that the proposed models of the RA can orient the definition of missions, emergent behaviors, objects and capabilities of constituent of HSH systems. However, more scenarios can be used to have more evidence and to identify possible omissions in the RA models presented in this paper.

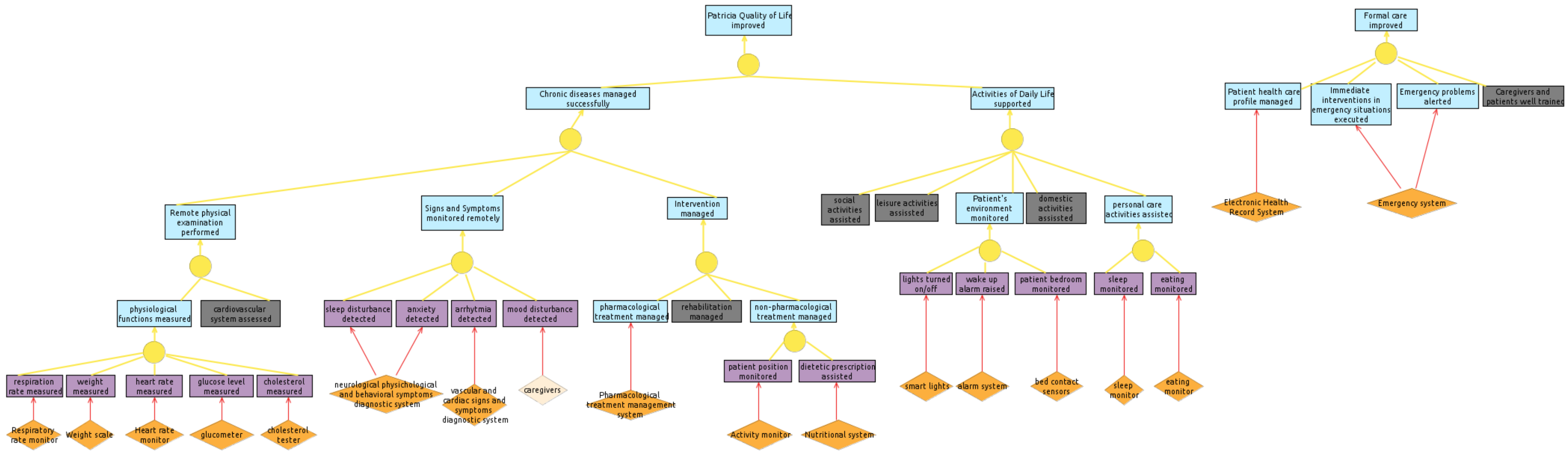


Figure 3.19: Missions model for a concrete HSH system

.

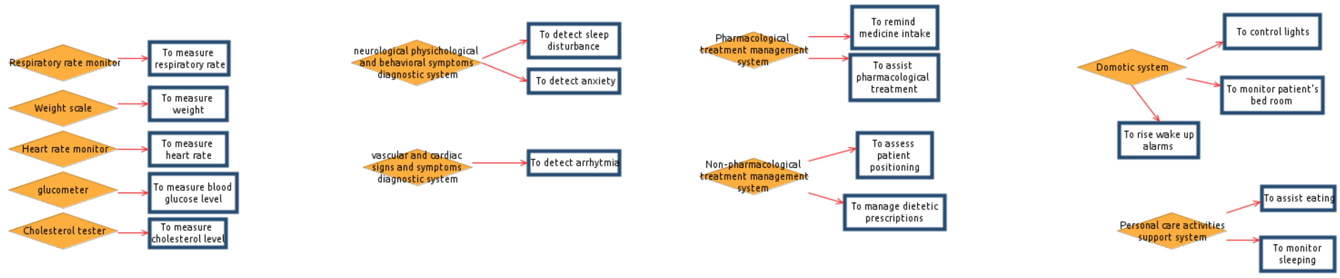


Figure 3.20: Responsibilities model for a concrete HSH system

.

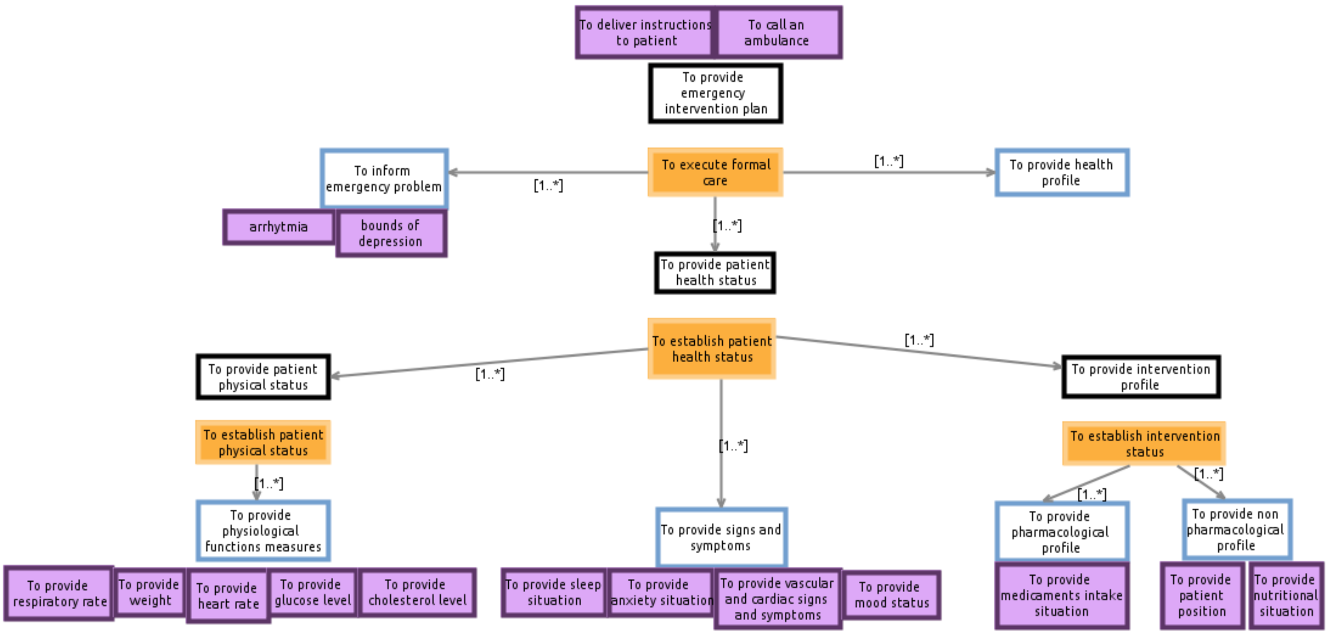


Figure 3.21: Emergent behaviors model for a concrete HSH system

.

Chapter 4

Final Remarks

The knowledge contained in the RA could help to orient the analysis, development, standardization and evolution of HSH systems, offering a broad repository of their possible missions and emergent behaviors regarding capabilities offered by their constituents. In this report we presented the process used to establish and validate HSH systems missions in a RA. As result, we observed that models provided by the RA, can be reused to identify missions, emergent behaviors, objects (i.e., entities and events), constituent capabilities and responsibilities of HSH systems. Hence, such process helps to reduce efforts and time at early stages of concrete HSH systems development.

As future works we intend to develop more systematic guidelines based on model-driven engineering techniques to instantiate RA models to create concrete SoS specifications.

Acknowledgment

Authors thank to Carla Gómez, clinical nurse specialist in chronic diseases, for orienting the selection of clinical guidelines. This work is supported by the Brazilian funding agency FAPESP (Grants N.: 2015/19192-2, 2014/02244-7 and 2013/20317-9).

Bibliography

[1] Axel van Lamsweerde. Goal-Oriented Requirement Engineering: A Guided Tour. In 5th IEEE International Symposium on Requirements Engineering, 2001, pp. 249-263.

[2] Charles B. Keating, Jose J. Padilla, Kevin Adams. System of Systems Engineering Requirements: Challenges and Guidelines. Engineering Management Journal, vol. 20, No. 4, pp. 23-31, 2008.

[3] Claus Ballegaard Nielsen, Peter Gorm Larsen, John Fitzgerald, Jim Woodcok, Jan Peleska. System of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions. ACM Computing Surveys, vol. 48, No. 2, Article 18, pp. 1-41, 2015.

[4] David Beale and Joseph Bonometti. Chapter 2: Systems Engineering (SE). The Systems Design Process. In: ESMD Course Material : Fundamentals of Lunar and Systems Engineering for Senior Project Teams, with Application to a Lunar Excavator. Auburn University. Available in: `http://www.eng.auburn.edu/~dbeale/ESMDCourse/Chapter2.htm`. Accessed in June 28th, 2016.

[5] Eduardo Silva, Thais Batista, Everton Cavalcante. A Mission-Oriented Tool for System-of-Systems Modeling. In SESoS, 2015, pp. 31-36.

[6] Eduardo Silva, Thais Batista, Flavio Oquendo. A Mission-Oriented Approach for Designing System-of-Systems. In SoSE, 2015, pp. 346-351.

[7] Elisa Yumi Nakagawa and Lucas B. R. de Oliveira. Using Systematic Review to Elicit Requirements of Reference Architectures. In 14th WER, 2011, pp. 1-12.

[8] Elisa Yumi Nakagawa, Flavio Oquendo, Juan Carlos Maldonado. Reference Architectures. In: Oussalah, M. O., ed. *Software Architecture: Principles, Techniques, and Tools*, 1st ed, chap. 2, London: Wiley, pp. 55-82, 2014.

[9] Elisa Yumi Nakagawa, Flavio Oquendo, Paris Avgeriou, Carlos E. Cuesta, Khalil Drira, Jose Carlos Maldonado, Andrea Zisman. Foreword: Towards Reference Architectures for Systems-of-Systems. In SESoS 2015, pp. 1-4.

[10] Frank Wartena ; Philips Research Europe, Eindhoven, The Netherlands ; Johan Muskens ; Lars Schmitt ; Milan Petkovic. Continua: The reference architecture of a personal telehealth ecosystem. In Healthcom, 2010, pp. 1–6.

[11] Grace A. Lewis, Edwin Morris, Patrick Place, Soumya Simanta, Dennis B. Smith. Requirements Engineering for Systems of Systems. In IEEE SysCon, 2009, pp. 1-6.

[12] Henri Hietala, Veikko Ikonen, Ilkka Korhonen, Jaakko Lahteenmaki, Anna Maksimainen, Vesa Pakarinen, Juha Parkka, Niilo Saranummi. Feelgood - ecosystem of phr based products and services. Research report VTT-R-07000-09, VTT Technical Research Centre of Finland. pp. 1-69, 2009.

[13] Jo Ann Lane. System of Systems Capability to Requirements Engineering. In 9th SOSE, 2014, pp. 91-96.

[14] Lina Garcés, Apostolos Ampatzoglou, Paris Avgeriou, Elisa Yumi Nakagawa. A Reference Architecture for Healthcare Supportive Home Systems. In CBMS'15, 2015, pp. 22-25.

[15] Lina Garcés, Apostolos Ampatzoglou, Paris Avgeriou, Elisa Yumi Nakagawa. A Comparative Analysis of Reference Architectures for Healthcare in the Ambient Assisted Living Domain. In CBMS, 2015, pp. 270-275.

[16] Lina Garcés, Flavio Oquendo, Elisa Yumi Nakagawa. A Quality Model for Ambient Assisted Living Systems. In CBMS'16, 2016, pp. 1-6 (accepted to publish).

[17] Luis M. Camarinha-Matos, João Rosas, Filipa Ferrada, Ana Inês Oliveira, Hamideh Afsarmanesh, Miriam Brielmann (eds.). BRAID Project Booklet: ICT and Ageing Scenarios. pp. 1-37, 2010.

[18] Mark W. Maier. Architecting Principles for System-of-Systems. Systems Engineering, vol. 1, pp. 267-284, 1999.

[19] Michael Huch. BRAID Project. D2: Identification and characterisation of the main stakeholder groups for "ICT for Ageing" solutions. Version 5, pp. 1-46, 2010.

[20] Objectiver. A KAOS Tutorial. Version 1.0. 2007. `http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf`. Accessed in 15th June, 2016.

[21] OUSD(AT&L), DoD. Systems and Software Engineering. Systems Engineering Guide for Systems of Systems. Technical Report Version 1.0. Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Department of Defense. 2008.

[22] Patrick Heymans and Eric Dubois. Scenario-Based Techniques for Supporting the Elaboration and the Validation of Formal Requirements. Requirements Engineering Journal, vol. 3, No. 3-4, pp. 202-218, 1998.

[23] Queensland Goverment. Chronic Conditions Manual: Prevention and Management of Chronic Conditions in Australia. 1st edition, pp. 1-258, 2015. `https://publications.qld.gov.au/dataset/chronic-conditions-manual`. Accessed in 15th June, 2016.

[24] Randy G. Walker and Charles B. Keating. Defining SoS Requirements: An Early Glimpse at A Metholodogy. Int. J. System of Systems Engineering, vol. 3, No. 3-4, pp. 306-319, 2012.

[25] Robert Darimont and Axel van Lamsweerde. Formal Requirement Patterns for Goal-Driven Requirements Elaboration. In SIGSOFT, 1996, pp. 179-190.

[26] Samuil Angelov, Paul Grefen, Danny Greefhorst. A Framework for Analysis and Design of Software Reference Architectures. Information and Software Technology. vol. 54, 4. pp. 417-431, 2012.

[27] Sara Ercolani, Francis Real, Roberta Annicchiarico, Fabio Camapana, Alessia Federici, Patrizia Mecocci, David Riano. Knowledge-Based HomeCare eServices for an Ageing Europe. Deliverable D05.2: Case Profile Ontology. pp. 1-128, 2008.

[28] Saurabh Mittal and Larry Rainey. Harnessing Emergence: The Control and Design of Emergent Behavior in System of Systems Engineering. In SCSC, 2015, pp. 1-10.