A Testing and Certification Methodology for an Open Ambient-Assisted Living Ecosystem

João Pascoal Faria, *INESC TEC and Faculdade de Engenharia da Universidade do Porto, Portugal*

Bruno Lima, *INESC TEC and Faculdade de Engenharia da Universidade do Porto, Portugal*

Tiago Boldt Sousa, *INESC TEC and Faculdade de Engenharia da Universidade do Porto, Portugal*

Angelo Martins, *INESC TEC and Instituto Superior de Engenharia do Porto, Portugal*

ABSTRACT

To cope with the needs raised by the demographic changes in our society, several Ambient-Assisted Living (AAL) technologies have emerged in recent years, but those 'first offers' are often monolithic, incompatible and thus expensive and potentially not sustainable. The AAL4ALL project aims at improving that situation through the development of an open ecosystem of interoperable AAL components (products and services), tied together by an integration infrastructure, comprising a message-queue based service bus and gateways bridging the communication with devices. To that end, the project encompasses the specification of interfaces and requirements for interoperable components, against which candidates can be tested and certified before entering the ecosystem. This paper proposes a testing and certification methodology for such an ecosystem. Besides fulfilling specified pre-requisites, candidate components must pass unit tests that check their conformance with interface specifications and integration tests that check their semantic interoperability with other components in specified orchestration scenarios.

Keywords: *Testing; Certification; Ambient-Assisted Living; Ecosystem; Interoperability*

Introduction

Developed societies are currently facing severe demographic changes: the world is getting older at an unprecedented rate. According to the United Nations (DESA, 2009), in 1950 there were 205 million persons (8% of the world's population) aged 60 or over. By 2009, the number of persons aged 60 or over had increased 3.5 times to 737 million (11% of the worlds' population). By 2050 the number of persons aged 60 or over is projected to increase again nearly threefold to reach 2 billion (22% of the worlds' population). This demographic trend will be also followed by an increase of people with physical limitations. New challenges will be raised to the traditional systems of health care. Hence, there is an urgent need to find solutions that allow extending the time people can live in their preferred environment by increasing their autonomy, self-confidence and mobility.

In recent years, some technologies for Ambient-Assisted Living (AAL) (for example, TCare (TCare, 2013)) have emerged with the objective of enabling their users to live independently for a longer period of time, increasing their autonomy and confidence in performing everyday tasks. Age, illness, and permanent or temporary disabilities are frequent causes of loss of autonomy. AAL technologies allow better monitoring and care of the individual, enhancing their security and simultaneously reducing the varied resources and associated costs. However, existing AAL 'first offers' for primary and secondary end-users are often monolithic, incompatible and thus expensive and potentially not sustainable.

The AAL4ALL project (AAL4ALL, 2011) aims at answering to those problems through the development of an ecosystem of interoperable products and services for AAL, tied together by an integration infrastructure, with an associated business model, and validated through a large scale trial. The project joins more than 30 relevant stakeholders, like public institutions, industry, user organizations and R&D institutions. One goal of this project is to ensure that any supplier of AAL products and services, whether they are physical devices or software, can enter the ecosystem easily and independently, whilst assuring their interoperability with the rest of the ecosystem. To that end, the project encompasses the specification of standards and reference models for AAL products and services, against which candidate products and services can be certified and subsequently integrated as components of the ecosystem. Already existing standards are taken into account whenever applicable, in order to avoid reinventing the wheel and bring the focus only to the missing pieces. The project also encompasses the definition of a testing and certification methodology for such components, which is the subject of this paper.

The certification and corresponding labeling of products and services has several advantages relevant for the AAL ecosystem: assure their compliance with important technical regulations (for example for safety, compatibility, etc.); assist regulatory bodies in their market surveillance of labeled products covered by their responsibility; assist manufacturers in the selection of certified components for their own products and facilitate the subsequent certification of their own assembled products; help suppliers and retailers selling their products; give consumers a higher confidence on the products they use and buy; enable interoperability in a multi-vendor, multi-network, multi-service environment, regarding different categories (Veer & Wiles, 2008) , such as technical, syntactical, semantic and organizational interoperability.

The main contribution of this paper is the definition of a novel testing and certification methodology, which was developed for the AAL4ALL ecosystem, but can be applied for other open socio-technical service systems.

The rest of the paper is organized as follows: the second section presents the integration architecture defined for the AAL ecosystem; the third section presents the proposed standardization, testing and certification processes; the fourth and fifth sections present a metamodel with the main concepts involved as well as test coverage and certification criteria; the sixth section presents an example and preliminary results; related work is presented in the seventh section; conclusions an future work are drawn in the last section.

AAL Ecosystem Integration Architecture

Before presenting the testing and certification methodology for AAL products and services to be integrated in the ecosystem, it is important to first describe the integration architecture designed by the authors in cooperation with other project partners for the AAL ecosystem.

This project's major goal was to facilitate the integration of existing e-health systems in order to create an open ecosystem of e-health devices and services, where vendors could cooperate to reach a wider audience. Such required a scalable infrastructure so that partners could integrate their devices and services in the ecosystem, as well as a core set of ICT services to manage it.

The initial approach at designing such infrastructure consisted on the identification and categorization of the components to be integrated. Three major categories were identified: sensors, actuators and services. Sensors are devices that are either worn by patients, or that are deployed in the surroundings of their living environment. These capture information regarding his body or the environment itself; data captured might range from the heart rate to room temperatures or if a door is closed. They provide data that can be used to evaluate the patient's wellbeing. Actuators are similar to sensors, but with the ability to control a physical device. Examples are remote light switches or medicine dispensers, which can be instructed to perform some action at a given time. Finally, services are software components that receive and process data, finally making it available to stakeholders for decision making or other purposes. Purely functional services might exist, gathering and transforming information so that others have it transformed into the input type they are expecting. Interactive services may also receive inputs from stakeholders and transmit them to the appropriate destinations (devices or other services).

The second phase consisted on identifying communication patterns to adopt as ecosystem's standards. The authors quickly understood that a service bus based on message queues would be the best approach to propagate messages between components (Schramm, Kostinger, Bayrhammer, Fiedler, & Grechenig, 2012), as these would be widely distributed across the Internet, with possible unreliable connections, mainly in mobile scenarios. The publisher-subscriber pattern (Schmidt & O'Ryan, 2003) could be adopted, by having components publish their data to the ecosystem and consuming data targeted at them, using a topic-based addressing scheme, in a weakly coupled, asynchronous and scalable fashion. This way, each component would be a publisher and/or subscriber in the ecosystem. On the other hand, security is a major concern while designing the infrastructure, as most data contains sensitive information regarding patients, which should only be delivered to authorized targets. For that, messages may circulate encrypted in the ecosystem whenever needed, and each component in the ecosystem is given appropriate authentication credentials with an associated set of authorized topics for publication and subscription. Such credentials and permissions are managed by a *registry* ICT service, handled by the ecosystem's administrators. Since a more powerful content-based addressing scheme, with subscriptions based not only on message topics but also on message contents, would conflict with privacy and encryption requirements, topics may have a structure as rich as needed for addressing and routing purposes, acting as routing keys. Whenever needed, a routing key may have an associated pair of public and private keys, to be used by sources and targets, respectively, in message encryption and decryption. Another major concern is reliability and fault tolerance in message delivery. For that, a *caching* ICT service should be able to temporarily store messages for durable subscribers while they are offline, and deliver them when they come back online.

The following phase consisted on the implementation of the system, requiring the decision of which technologies would better solve the problem at hand, while fulfilling all requirements. Considering the need to create a service bus with publisher-subscriber capabilities based on routing keys, AMQP (Fernandes, Lopes, Rodrigues, & Ullah, 2013) revealed the most apt protocol to adopt. However, publication to the ecosystem required validation, resulting in the need to proxy data publication. To circumvent this issue, we adopted HTTP as the protocol used by publishers to send information into the ecosystem. This first layer validates all incoming information, regarding syntax and permissions, only then injecting it into the AMQP bus, with the proper routing keys and sender information. The AMQP protocol will then replicate the message by all queues subscribed to the given routing keys. To this system we have given the name of AALMQ, for Ambient Assisted Living Message Queue. While performing integration tests with partners, the need to scale dynamically was identified, as components would join or leave the ecosystem at any given time. To provide a scaling mechanism, the receiving point, which uses HTTP, was built as being stateless and able to use all available CPU cores, meaning that it can be scaled by itself, both locally or horizontally, without synchronization of nodes. Only required is a valid address for where the AMQP service is running, which can be a remote address. As for AMQP, the protocol facilitates the creation of high-availability clusters. More machines could be added to the initial nodes in runtime. At both points, a load-balancer would be adopted to split load between nodes.

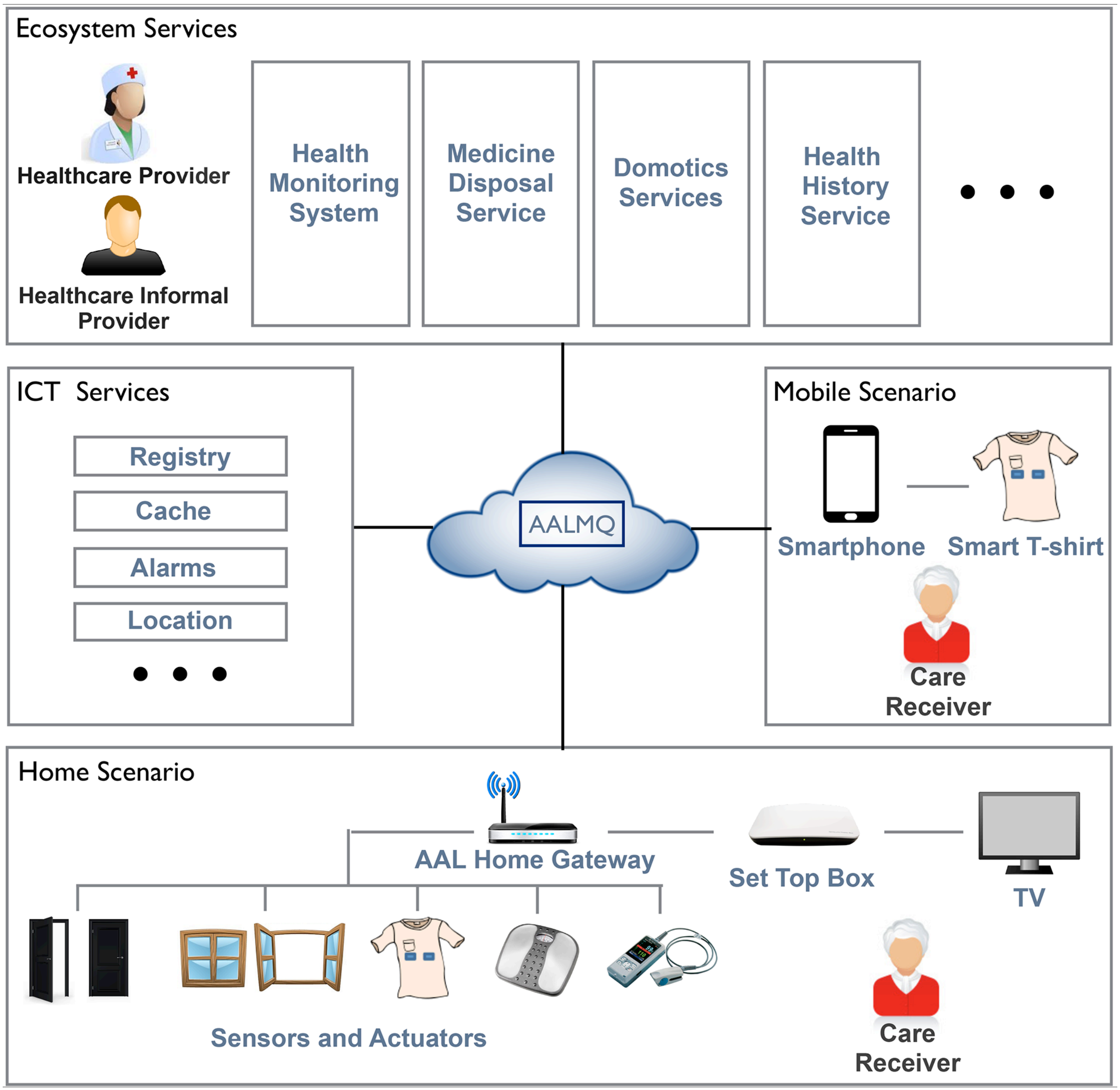

Figure 1. AAL4ALL Architecture

An additional component was required to create the AAL4ALL ecosystem: gateway devices or services. These would integrate with AALMQ as any other components, but would facilitate the deployment of a multitude of devices with patients. Gateways bridge the communication between devices with limited protocol capabilities, such as a heart rate Bluetooth sensor, and the ecosystem. After configuration, these load the required drivers to communicate with nearby sensors and actuators, transforming data from the format supported by the device to a standard adopted in the ecosystem, such as the ISO 11073 (Jianchu & Warren, 2005). After specifying the role of gateways in the ecosystem, any partner could implement his own gateway and have it support as much devices as seem fit. Also, gateways could be deployed in the cloud to be used as a bridge to communicate with mobile phones that would act or communicate themselves with sensors and actuators. This allowed to reduce the computation required by these devices and also to prevent possible connection problems from bad network connections, acting as a cache between them and the ecosystem.

This publish-subscribe based approach allowed the simplified composition of services, meaning that workflows could be defined where a service's output is another's input. Also, the same routing key could be subscribed by multiple ecosystem components. A complete orchestration scenario could consist on the capture of health information with the patient, such as what has been done in the CAALYX project (Prescher et al., 2012), with the gateway propagating information to the ecosystem. A service for processing alarms could evaluate all acquired information (Ferreira, Sousa, & Martins, 2012) with alarms being sent either directly to caretakers via SMS or email, or re-sent into the ecosystem to be shown by yet another service that consumes them, providing an interface for the patient's caretakers or himself to monitor his status.

As seen from Figure 1, the AALMQ is responsible for distributing information between all components in the ecosystem. From a physical point of view, integrated components can be from one of the following categories. ICT Services are services that value the ecosystem by providing partners with a core set of services which they can rely on to build their own services, such as Cache or Alarm detection and delivery. The Registry service is directly responsible for registering components in the ecosystem and managing their communication permissions, as well as passing those to AALMQ, making it an essential service in the ecosystem. At the patient's home, a router/gateway bridges the communication between local devices and the ecosystem, managing multi-protocol connection to all existing devices. Mobile scenarios are supported and happen when the patient leaves his home carrying one or more sensors with him. All data is acquired with a mobile device that uses a data plan connection, via 3G, EDGE or similar, to establish a network connection with AALMQ and the ecosystem. Some mobile scenarios might communicate with a centralizing gateway that bridges communication between mobile devices and the ecosystem just as the home gateway does. Finally, the ecosystem services is where services from partners exist, each deployed in their own cloud infrastructure, communicating with the remaining components through the AALMQ.

Standardization, Testing and Certification Process

The standardization, testing and certification activities are to be carried out by the two main actors defined in the business model for the AAL4ALL project -- the AAL4ALL Association and the AAL4ALL Certification Body (AAL4ALL, 2013). Their missions and responsibilities are described next, emphasizing the ones related with the scope of this paper.

Actors and responsibilities

The AAL4ALL Association plays the role of a standardization authority. Being currently represented by the project consortium, it should in the future integrate all interested stakeholders of the AAL ecosystem, such as product manufacturers and vendors, service providers, end-user associations, R&D institutions, public administration entities, sectorial clusters (such as the Health Cluster Portugal consortium (Health Cluster Portugal, 2013)), among others. The nonprofit AAL4ALL Association should be a regulatory and proactive entity of the AAL4ALL ecosystem, with the following responsibilities more closely related with standardization, testing and certification:

- Development and maintenance of standards and reference models for AAL products and services, possibly in cooperation with national standardization authorities. For that purpose, the association may appoint technical commissions responsible to prepare relevant materials to be subsequently approved by the AAL4ALL Association;
- Definition of test and certification criteria to be followed by AAL4ALL Certification Bodies in test case design and certification;

- Accreditation or recognition of one or more AAL4ALL Certification Bodies, taking into account technical competence and independence;
- Overseeing the entities responsible for managing the AAL4ALL infrastructure;
- Promotion of AAL4ALL standards, reference models, labels and ecosystem;
- Contribution to the development of international standards and reference models in the AAL domain;
- Appointment of a consumer ombudsman for the AAL4ALL ecosystem.

The sustainability of the AAL4ALL Association will be ensured by multiple revenue sources: a fee levied on each member; a fee levied on each AAL4ALL certificate issued; organization of thematic events; participation in co-financed projects related with its mission and responsibilities.

On the other hand, the AAL4ALL Certification Body plays the role of a certification entity. Being currently represented by project partners responsible for the testing and certification activities, it may in the future take the form of an association that subcontracts the testing and certification activities to its associates. In order to avoid conflicts of interests, manufacturers or vendors of AAL products or services candidate for certification cannot be part of the AAL4ALL Certification Body.

The main responsibilities of the AAL4ALL Certification Body are:

- Prepare generic test plans (including test procedures and test cases) to assess the conformity of candidate products and services with respect to the certification requirements specified in reference models and standards issued by the AAL4ALL Association, according to the test coverage and certification criteria defined by the AAL4ALL Association;
- Develop appropriate testing infrastructures, or otherwise establish partnerships with properly equipped test laboratories from its associates or third parties;
- Tailor and instantiate the generic test plans for each candidate product or service;
- Conduct conformance tests on candidate products and services, based on defined test plans for the product or service at hand, or otherwise subcontract those activities to test laboratories from third parties or from its associates;
- Issue AAL4ALL certificates and labels for the products and services that passed the tests and other assessment criteria, or non-conformance reports in the opposite case;
- Publicize the AAL4ALL label and certificates issued;
- Manage the lifecycle of certificates (issuing, renewal, upgrade, revocation, etc.) according to policies mutually agreed upon with the AAL4ALL Association.

This will be a self-sustained entity that generates revenues primarily through the fees associated with the testing and certification activities and certificate lifecycle management.

Finally, interested product manufactures are responsible to send AAL4ALL certification requests to an accredited AAL4ALL Certification Body. Product manufactures should have prior access to the relevant standards, reference models, certification criteria, and generic test plans, in order to properly prepare their products for certification. Since a certificate is issued for a specific product version, product manufactures should request from the AAL4ALL Certification Body certificate upgrades for new versions of their products, which may be issued following simplified procedures.

All the actors mentioned above, namely the AAL4ALL Association, the AAL4ALL Certification Body and test laboratories shall be subject to the requirements defined by the ISO/IEC 17011:2004 standard (ISO/IEC, 2004), ISO/IEC 17065:2012 standard (ISO/IEC, 2012a) and ISO/IEC 17025:2005 standard (ISO/IEC, 2005), respectively.

Activities and artifacts

Having described the main actors and responsibilities, we next describe the main activities (steps) and artifacts involved in the process. Figure 2 presents an overview of the overall standardization, testing and certification process conceived for the ecosystem.

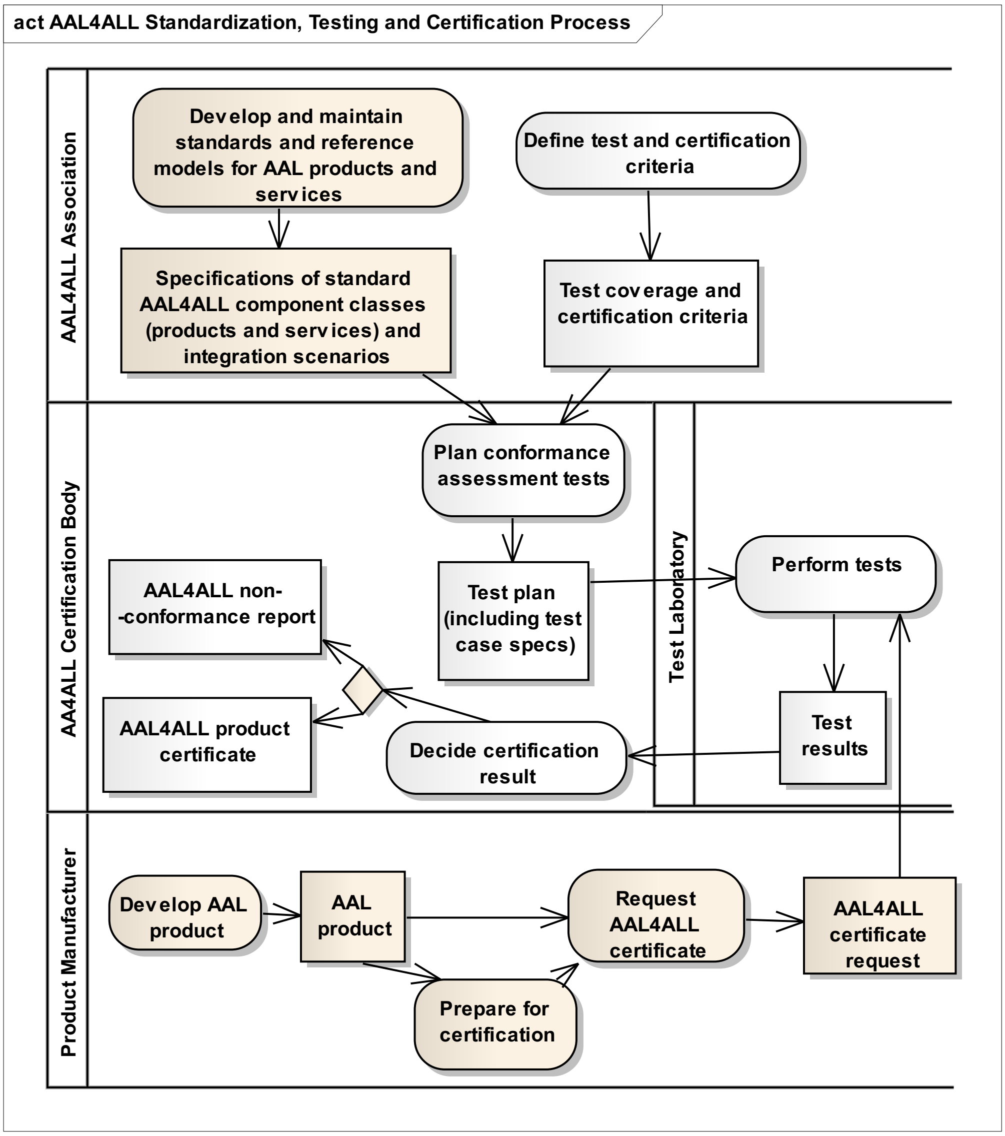

Figure 2. UML activity diagram describing the overall AAL4ALL standardization, testing and certification process

The testing and certification process for a candidate component begins always with the request of a Product Manufacturer of a product for which he wants to obtain the AAL4ALL label. Upon receipt of such request, the AAL4ALL Certification Body analyzes it and prepares a test plan (possibly instantiating a generic test plan previously prepared), to be followed by the chosen test laboratory. When the manufacturer sends the product to the Test Laboratory, the Test Laboratory conducts the testing and, in the end, communicates the results to the AAL4ALL Certification Body. The AAL4ALL Certification Body will then review those results and determine whether the product has completed the certification process successfully. If so, the product receives an AAL4ALL product certificate, otherwise, the Product Manufacturer receives a non-conformance report specifying the problems identified.

However there are also cases that do not follow the generic process outlined above, particularly in the case of a renewal of certification or the upgrading to a new product version. In these cases the process will be more simple and fast, but with the same rigorous review.

The diagram of Figure 2 also depicts the major standardization and test planning activities that are under the responsibility of the AAL4ALL Association and the AAL4ALL Certification Body, as described in the previous sub-section.

Metamodel for the Specification of Components and Test Cases

The metamodel depicted in Figure 3 presents an overview of the concepts involved in the specification of standard AAL component classes, integration scenarios, and test cases. We use the term *metamodel* because the specifications themselves may be regarded as component models, interaction models and test models, respectively. The presented metamodel served as a basis for the definition of specification templates as illustrated in a later section.

Components

Regarding the specification of standard AAL component classes, the central concept is that of a component. A component may be of one the following kinds: device (product) or software application (software service). In the metamodel of Figure 3, the meta-class *Component* represents a component class (e.g., AAL Sensor, AAL Actuator, AALMQ, AAL Gateway, AAL Registry Service, AAL Monitoring App, etc.), and not a particular implementation (e.g., a particular implementation of an AAL Gateway by a particular manufacturer), or a particular instance of that implementation (e.g., a gateway with a serial number).

To ensure syntactic and semantic interoperability within the ecosystem, components should communicate with other components in the ecosystem through well-defined (standardized) interfaces, following well-defined (standardized) contracts for message sending and reception. However, communications with components outside the ecosystem need not be standardized. For each component class, it is specified a set of interfaces that components in that class should support (software interfaces, hardware interfaces or user interfaces), the messages they should be able to send and receive through those interfaces, the events that trigger the sending of messages (e.g., receiving a message, reaching an internal state, or a time event), the effects (or post-conditions) of messages received (e.g., changing internal state, or sending a message), the conditions (or pre-conditions) upon which messages may be received (e.g., being in some internal state). In order to facilitate the formalization of the contracts for message sending and receiving, abstract state variables may be used.

In the metamodel of Figure 3, the meta-class *Message* represents a message type (e.g*., Publish*), and not a particular usage in a sequence diagram or a particular occurrence at run-time (with a particular timestamp, field values, etc.). Messages are described by their name, fields and constraints on field values. Messages are composable, i.e., they may have other messages as fields.

In order to allow keeping component specifications as generic and abstract as possible, the possible realization options (e.g., concrete communication protocols) are described separately.

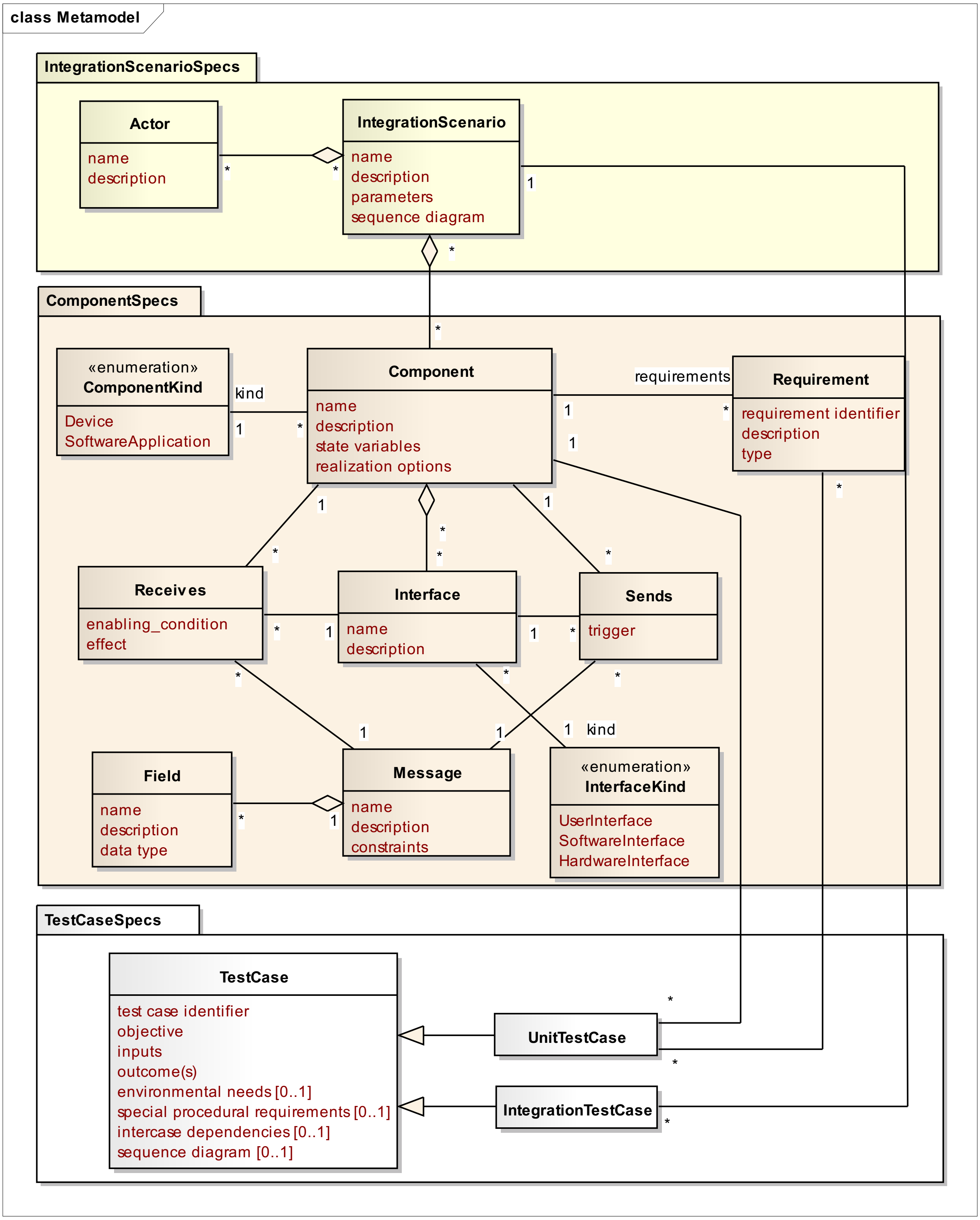

Figure 3. UML class diagram partially describing the metamodel for the specification of standard AAL components and associated test cases

For each component class, it may also be specified a set of supplementary requirements that components in that class must fulfill (e.g., security requirements or performance requirements). Some of these requirements may correspond to prerequisites (e.g., CE labeling, compliance with ISO/IEEE 11073 standard (ISO/IEEE, 2004) that are to be checked prior to testing. For the sake of completeness, functional requirements underlying the defined interfaces and contracts may also be specified. Requirements have a unique identifier, for traceability purposes, and a type (Security, Usability, etc.) to be assigned based on the ISO/IEC 25010:2011 standard (ISO/IEC, 2011).

Integration scenarios

In order to guide the design of integration test cases, it is also useful to specify a set of key integration scenarios, involving the orchestration of multiple components to provide end-to-end services to ecosystem actors (such as care providers and care receivers). Each of those scenarios (see meta-class *IntegrationScenario* in Figure 3), involving a set of actors and components, is described by a UML sequence diagram. For generality and readability, scenarios may be parameterized. Messages illustrated in a scenario should be consistent with the specification of the participating components.

Whilst specifications of component classes play a normative role, the specifications of integration scenarios play an informative role, to guide the design of integration test cases.

Test cases

The last part of the metamodel refers to the specification of test cases. Test cases are specified according to the IEEE Std 829™-2008 (IEEE Standards Association, 2008). Optionally, a sequence diagram may be provided to visually depict the components and interactions involved in the test. When meaningful, test cases may be associated with specific requirements, for traceability purposes.

Two test levels (Burnstein, 2003) are relevant in our context:

- unit (component) test cases, derived from the components' specifications according to coverage criteria to be presented in the next section;
- integration (scenario) test cases, derived from the integration scenarios' specifications according to coverage criteria to be presented in the next section.

Test Coverage and Certification Criteria

For each component class, a set of abstract unit test cases should be derived, covering all the messages, effects, conditions, triggers and requirements. During the testing process of a candidate component, the unit test cases defined for the component class may be refined taking into consideration the realization options of that component (such as protocols used, optional features supported, details of the user interface, etc.).

A set of abstract integration test cases should also be derived from each integration scenario, covering all the messages and branches. During the testing of a candidate component, a set of integration test cases in which the component class participates should be selected (based on defined criteria) and refined for that component.

To be successfully certified, a candidate component must pass the following verification steps:

1. Checking the fulfillment of the prerequisites specified for the component class, via the analysis of appropriate evidences.
2. Passing the units test cases derived from the component class specification.
3. Passing the integration test cases derived from a set of integration scenarios in which the component class participates, using certified components for the other participants.

Example and Preliminary Results

To better understand how the concepts presented can be instantiated, we present a concrete example for an AAL monitoring service. As illustrated in Figure 4, the scenario consists of a *Care Receiver* who has two sensors (*Scale* and *ECG*) and a networking device (*AAL Home Gateway*) at home. Somewhere in the cloud, an *AALMQ* node assures the communication between the *AAL Home Gateway* and the *Monitoring Web App* and *Monitoring Mobile App* used by a *Healthcare Provider* and a *Healthcare Informal Provider*, respectively. The *Healthcare Provider* represents an entity (doctor, nurse, group or organization) that is responsible for monitoring and responding to any problems that may arise with the *Care Receiver*, while the *Healthcare Informal Provider* is a non-specialized person (family, friend) that wants to be informed about the status of the *Care Receiver*.

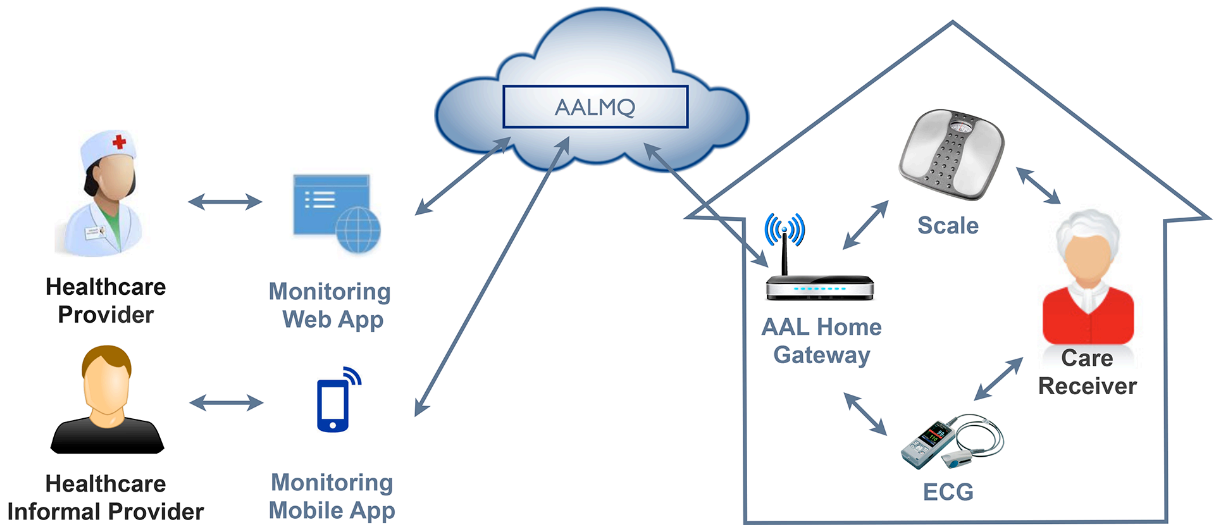

Figure 4. Example configuration for an AAL monitoring service

In this example, it is possible to identify different communication scenarios, one of which is illustrated in Figure 5. This scenario describes the mechanism for assuring that a *Healthcare Provider* (*p*) receives an alert when the heart rhythm of a *Care Receiver* (*r*) is outside a specified range (*min*, *max*). To simplify the description, it is assumed that the system has been previously configured and the participants are already able to communicate with each other. The scenario works as follows:

- The *Healthcare Provider* (*p*) starts by interacting with the *Monitoring Web App,* through its user interface, to request the monitoring of the heart rhythm signal of the *Care Receiver (r*) with a specific time period (*t*) and control range (*min*, *max*). This interaction is abstractly represented by the *MonitorSignal* message in Figure 5.
- The *Monitoring Web App* then prepares a *RequestData* message to be sent to the *AAL Gateway* that serves the *Care Receiver*, having as parameters the type of signal, the identification of the *Care Provider* requesting the data, the identification of the *Care Receiver* to be monitored, and the time period for data collection. The *Monitoring Web App* transmits the message via the *AALMQ* node, by sending a *Publish* message to the *AALMQ* node, having as parameters a routing key and the *RequestData* message. In this case, the routing key is a topic that identifies the target of the message, that is, the *Care Receiver*. If wanted, the *RequestData* message may be encrypted by the *Monitoring Web App*, becoming completely opaque to the *AALMQ* node.

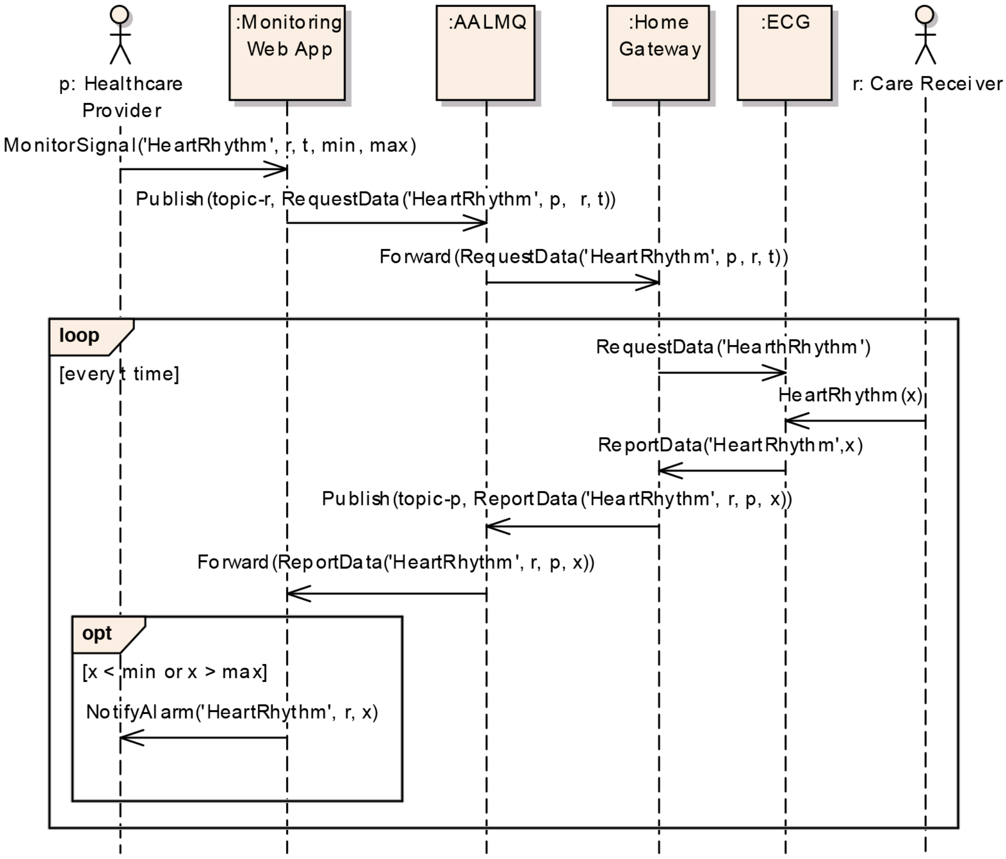

Figure 5. UML sequence diagram for an integration scenario S1

- Assuming that the indicated *AAL Home Gateway* previously subscribed the topic that identifies the *Care Receiver r*, the *AALMQ* node forwards the *RequestData* message to the *AAL Gateway*.
- The *AAL Home Gateway* then decrypts (if needed) and interprets the received message. Based on configuration information, it first determines the device that measures the requested signal (hearth rhythm) from the requested person (*r*). Then, as indicated by the *loop* interaction operator in Figure 5, it periodically requests (with time period *t*) a reading from the device*,* which reports back the measured value (*x*). For each value reported by the device, the gateway prepares a *ReportData* message to be sent to the requester, having as parameters the type of signal, the identification of the *Care Receiver* being monitored (*r*), the identification of the *Care Provider* that requested the data (*p*), and the actual measurement (*x*). The gateway transmits the message via the *AALMQ* node, by sending a *Publish* message having as parameters a routing key and the *ReportData* message. Like in the previous case, the routing key is a topic that identifies the target of the message, which in this case is the *Care Provider*. Again, if wanted, the *ReportData* message may go encrypted, becoming completely opaque to the *AALMQ* node.
- Assuming that the *Monitoring Web App* previously subscribed the topic that identifies *Care Provider p*, the *AALMQ* node forwards the *ReportData* message to the *Monitoring Web App*.
- The *Monitoring Web App* then decrypts (if needed) and interprets the received message. It appends the received value to its internal database for subsequent consultation and, in case it lays

outside the specified range, sends an alarm notification message to the *Care Provider*, indicating the type of signal, the identification of the *Care Receiver* (*r*) and the observed value (*x*).

An excerpt of a sample component specification is shown in Table 1. Signatures are described in UML (ISO/IEC, 2012b) and OCL notation (Clark & Warmer, 2002) (ISO/IEC, 2012c). Examples of test cases derived from the component specification and the integration scenario are presented in Table 2 and Table 3. All the templates used are based on the metamodel of Figure 4.

Table 1. Example of a component specification

Name	AALMQ
Description	Provide ecosystem integration and service orchestration capabilities, acquiring data from components and forwarding it to the proper destinations, according to the publish-subscribe model.
Kind	SoftwareApplication
Interfaces	name: PublisherInterface description: Interface that receives publication requests. kind: SoftwareInterface
	name: SubscriberInterface description: Interface that receives subscription requests. kind: SoftwareInterface
	name: CallbackInterface description: Interface to be implemented by subscribers, so that the AALMQ can forward published messages. kind: SoftwareInterface
State variables	SubscriptionsTable: Set(Tuple(topic: String, subscriberId: ID)).
Receives	message: Subscribe(subscriberId: ID, topics: Set(String)) interface: SubscriberInterface enabling cond.: always enabled effect: update SubscriptionsTable
	message: Publish(topic: String, message: Message) interface: PublisherInterface enabling cond.: always enabled effect: send Forward(message) to all subscribers of this topic.
Sends	message: Forward(topic: String, message: Message) interface: CallbackInterface trigger: see Publish message
Realization options	Ethernet, AMQP (for interface SubscriberInterface and CallbackInterface) and HTTP (PublisherInterface)
Requirements	requirement id.: AALMQ.R1 description: The AALMQ should forward messages published on a topic to all subscribers of that topic. type: Functional Suitability
	requirement id.: AALMQ.R2 description: Message sending delay should be less than 200 ms. type: Performance Efficiency
	requirement id.: AALMQ.R3 description: Publishers and subscribers have to authenticate themselves using the credentials given as part of the certification process. type: Security

Table 2. Partial example of a unit test case specification

Test case identifier	AALMQ.T1
Objective	Check if a message (belonging to a particular topic) is received by a subscriber.
Inputs	Subscribe(id1, Set{topic1}) Publish(topic1 , msg1) (id1, msg1 and topic1 are test parameters to be defined in concrete test cases)
Outcome(s)	Subscriber 1 receives msg1
Requirements' identifiers	AALMQ.R1
Environmental needs	-
Special procedural requirements	-
Intercase dependencies	-
Sequence diagram	-

Table 3. Partial example of integration test case specification

Test case identifier	S1.1
Objective	Exercise a candidate component in scenario S1, and check the correct end-to-end behavior from the actors' perspective, covering all the scenario messages and conditions.
Inputs	Form healthcare provider: Request heart rhythm alarms via the Monitoring Web App, regarding a given person *r*, with a certain periodicity *t*, for values outside specified *min* and *max* values. Form healthcare receiver *r*: provide heart rhythm information via attached ECG device.
Outcome(s)	To healthcare provider: Alarm reports whenever appropriate.
Environmental needs	Devices, applications and actors needed to fulfill the scenario, plus a information recorder near the monitored person. Depending on the kind of ECG used, a separate device may be needed to show the user heart rhythm.
Special procedural requirements	Healthcare receiver should alternate between rest periods with low heart rate and high activity periods with high heart rate (e.g., running). Experiment with different time periods (*t*), and heart rate ranges (*min* and *max*). Compare values (*x*) collected near the user and the Web App. All but the candidate component should be certified components (with the goal of testing the non-certified component for this scenario).
Intercase dependencies	-
Sequence diagram	(see Figure 5)

The example presented is based on a small pilot case that was conducted to validate and refine the testing and certification approach presented in this paper. A testing infrastructure, comprising reusable test drivers and stubs, was developed to facilitate the implementation and execution of unit test cases.

Because some non-functional requirements were not fulfilled by the components under test in this phase of the AAL4ALL project, non-conformance reports where produced. We are currently conducting a larger pilot case, to further validate and refine the approach.

Related Work

In order to try to solve the problems described at the beginning of this paper, the emergence of projects concerned with the standardization and interoperability in the eHealth area has been growing. Some of these projects arise not only through consortia of companies that produce products for the health sector, but also by institutions such as the European Union.

The Continua Health Alliance (Continua Health Alliance, 2013) is a nonprofit, open industry alliance of healthcare and technology companies in the world joining together in collaboration to improve the quality of personal healthcare. Through the efforts of a collaborative industry organization, Continua aims at enabling a personal health ecosystem where many diverse vendors can combine their products into new value propositions with significant health benefits for people worldwide. One of the differences with the AAL4ALL project is that the Continua Health Alliance is focused on user devices and home communication, while the AAL4ALL project scopes end-to-end services, including also the communication with applications from the caretaker side, which interfaces are also subject to certification, and non-health related devices. The other difference is that the AAL4ALL project also includes the creation of an open integration infrastructure, enabling the interoperation of certified components, information sharing, service composition, and easy addition of new information providers and consumers. Of course, Continua standards may be used as prerequisites in our project whenever applicable.

EHR-Q^{TN} (EuroRec, 2013) is a Thematic Network project that prepares the health community across Europe for systematic and comparable quality assurance and certification of e-Health products, more specifically of Electronic Healthcare Record (EHR) systems. Although having a different focus, the EHR-Q^{TN} project produced a set of recommendations that may be applied also for the AAL ecosystem regarding the quality labeling and certification procedures, namely, the recommendations for third party assessments, start small, and incentivized model.

The Healthcare Interoperability Testing and Conformance Harmonization (HITCH) project (HITCH Project, 2011b) developed in the 2010-2011 period, produced a set of recommendations to the EU Commission on how to proceed with eHealth interoperability testing and certification/labeling in Europe. A major recommendation is the development of a common European testing and certification foundation for interoperability in healthcare systems, upon which national or regional eHealth projects across Europe may organize their own testing and certification procedures in the future (HITCH Project, 2011a). Although such a common foundation could be very useful for our project, as far as we know, it was not yet created.

Conclusions and Future Work

It was presented an approach for the standardization, testing and certification of interoperable products and services for an AAL ecosystem that is being developed in the context of a nation-wide project involving a large number of partners from different areas (sensor manufacturers, academic institutions, telecommunications operators, etc.).

The approach is currently being applied and validated for real-world pilot scenarios, combining products developed in the scope of the AAL4ALL project together with products that already existed in the market, thus allowing that the validation is as real as possible. If successful, we expect that the results achieved can be expanded and replicated in other countries worldwide.

As future work, we intend to create a test laboratory to automate unit and integration tests of candidate products (especially products that have software interfaces) and hence expedite the test and certification process.

Acknowledgement

This work is financed by the ERDF – European Regional Development Fund through the COMPETE programme (operational programme for competitiveness) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within projects AAL4ALL - «FCOMP-01-0202-FEDER-013852» (details at www.aal4all.org) and «FCOMP - 01-0124-FEDER-022701».

References

AAL4ALL. (2011). Ambient Assisted Living for All. Retrieved June, 2013, from http://www.aal4all.org/

AAL4ALL. (2013). *Definição de Modelo de Negócio e Gestão da Actividade*.

Burnstein, I.. (2003). *Practical Software Testing: A Process-Oriented Approach*. Springer.

Clark, T., & Warmer, J. (Eds.) (2002). *Object Modeling with the OCL: The Rationale behind the Object Constraint Language*. Springer.

Continua Health Alliance. (2013). Continua health alliance web site. Retrieved March, 2014, from http://www.continuaalliance.org/

DESA. (2009). World Population Ageing 2009. United Nations.

EuroRec. (2013). Thematic Network on Quality Labelling and Certification of EHR Systems. Retrieved March, 2014, from http://www.eurorec.org/rd/index.cfm

Fernandes, J. L., Lopes, I. C., Rodrigues, J. J. P. C., & Ullah, S. (2013). Performance evaluation of RESTful web services and AMQP protocol. In *2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN)* (pp. 810-815). IEEE Communications Society.

Ferreira, H. S., Sousa, T. B., & Angelo, M. (2012). Scalable Integration of Multiple Health Sensor Data for Observing Medical Patterns. In Y. Luo (Ed.), *Cooperative Design, Visualization, and Engineering* (pp. 78-84).

Health Cluster Portugal. (2013). From knowledge to market — Health Cluster Portugal. from http://www.healthportugal.com/

HITCH Project. (2011a). *HITCH Key Messages*. Retrieved from http://www.hitch-project.eu/sites/www.hitch-project.eu/files/HITCH-Key-Messages.pdf

HITCH Project. (2011b). Welcome to HITCH Building the Road to eHealth Interoperability!, Retrieved March, 2014, from http://www.hitch-project.eu/

IEEE Standards Association. (2008). IEEE Standard for Software and System Test Documentation 829-2008.

ISO/IEC. (2004). ISO/IEC 17011:2004 Conformity assessment -- General requirements for accreditation bodies accrediting conformity assessment bodies.

ISO/IEC. (2005). ISO/IEC 17025:2005 General requirements for the competence of testing and calibration laboratories.

ISO/IEEE. (2004). ISO/IEEE 11073-20101:2004 Health informatics -- Point-of-care medical device communication -- Part 20101: Application profiles -- Base standard.

ISO/IEC. (2011). ISO/IEC 25010:2011 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models.

ISO/IEC. (2012a). ISO/IEC 17065:2012 Conformity assessment -- Requirements for bodies certifying products, processes and services.

ISO/IEC. (2012b). ISO/IEC 19505-2:2012(E) Information technology - Object Management Group - Unified Modeling Language (OMG UML), Superstructure. Retrieved March, 2014, from http://www.omg.org/spec/UML/ISO/19505-2/PDF/

ISO/IEC. (2012c). ISO/IEC 19507:2012(E) Information technology - Object Management Group Object Constraint Language (OCL). Retrieved March, 2014, from http://www.omg.org/spec/OCL/ISO/19507/PDF/

Jianchu, Y., & Warren, S. (2005). Applying the ISO/IEEE 11073 Standards to Wearable Home Health Monitoring Systems. *Journal of Clinical Monitoring and Computing, 19(6)* (pp. 427-436). Springer.

Prescher, S., Bourke, A. K., Koehler, F., Martins, A., Ferreira, H. F., Sousa, T. B., & Nelson, J. (2012). Ubiquitous ambient assisted living solution to promote safer independent living in older adults suffering from co-morbidity. In *Proceedings of the 34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'12)*, San Diego, CA.

Schmidt, D. C., & O'Ryan, C. (2003). Patterns and performance of distributed real-time and embedded publisher/subscriber architectures. *Journal of Systems and Software - Special issue on: Software architecture - Engineering quality attributes, 66(3)* (pp. 213-223), New York, NY: Elsevier Science Inc.

Schramm, W., Kostinger, H., Bayrhammer, K., Fiedler, M., & Grechenig, T. (2012). Developing a hospital information system ecosystem for creating new clinical collaboration methodologies. In *2012 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)* (pp. 101-103), Hong Kong.

TCare. (2013). TCare - Conhecimento e Saúde S.A. Retrieved June, 2013, from http://www.tcare.pt/

Veer, H., & Wiles, A. (2008). Achieving Technical Interoperability – The ETSI Approach. European Telecommunications Standards Institute. Retrieved March, 2014, from http://www.etsi.org/WebSite/document/whitepapers/IOP%20whitepaper%20Edition%203%20final.pdf