A Catalog of Interoperability Solutions for Ambient Assisted Living

Paulo Duarte
Federal University of Ceara (UFC)
Fortaleza – CE, Brazil
pauloduarte@great.ufc.br

Rainara Carvalho
Federal University of Ceara (UFC)
Quixada – CE, Brazil
rainara@ufc.br

Windson Viana
Federal University of Ceara (UFC)
Fortaleza – CE, Brazil
windson@virtual.ufc.br

Abstract

As the global population of older adults continues to increase, there is a growing need for innovative technology solutions to improve their well-being and care. One prominent example is the Ambient-Assisted Living (AAL) domain, which involves the connection of many subsystems and heterogeneous devices to offer intelligent services in the user's living ambient. One of the main challenges in developing AAL systems is the lack of interoperability, which can occur at different levels (from integration between different systems to the semantic level of the data). Although there are several solutions and approaches to this problem, there is no organized body knowledge about interoperability for AAL systems impacting decision making on which approach to use in developing this type of system. This paper proposes a Non-Functional Requirements (NFR) catalog dedicated to interoperability in AAL systems. This catalog establishes relationships among technologic strategies, architectural patterns, platform types, communication protocols, and data semantic and syntactic aspects. We extracted data about interoperability in AAL systems from the academic literature. We identified 25 studies focused on AAL systems prioritizing interoperability as a critical requirement by conducting a systematic mapping followed by a forward snowballing process. We then employed Grounded Theory (GT) to extract information regarding the interoperability characteristics specific to AAL from these documents, finding 41 softgoals and 29 operationalizations. Finally, we validate our findings with two specialists.

Keywords

AAL, Interoperability, NFR Catalogs

1 Introduction

The older adult population worldwide is experiencing gradual growth. Recent studies indicate that by 2050, the proportion of older adults is projected to reach approximately 20-25% [35] [21]. This demographic shift will mark the first time the more senior population surpasses the younger population [27]. Consequently, it's expected to increase pressure on healthcare systems worldwide, which need to adapt to meet the specific needs of this age group, such as the treatment of chronic diseases and long-term care.

Numerous initiatives have emerged based on ambient intelligence principles focusing on improving users' well-being. One notable example is Ambient-Assisted Living (AAL) [35] [21]. According to the AAL Active and Assisted Living Programme [1], AAL is defined as "the use of Information and Communication Technologies (ICT) in a person's daily living and working environment to enable them to remain active, socially connected, and live independently into old age" [27]. AAL focuses on developing solutions for caring and monitoring older adults' daily lives. AAL solutions encompass a range of applications, from assistive robots [12] to smart homes equipped with environmental sensors [13].

One of the most common requirements in AAL system development is interoperability [10]. This is caused by the heterogeneity of devices, communication protocols, programming languages used by different manufacturers for purposes that should complement each other. Interoperability is not a recent problem. In fact, there are several mature solutions and approaches to solving it. However, interoperability can occur at many levels, from the infrastructure level (*e.g.*, communication between disparate systems) to the application level (*e.g.*, information exchange between applications using different protocols).

Nonetheless, the vast scope of interoperability presents a challenge with its myriad solutions, which, in turn, also becomes a problem. Although there are several solutions and approaches to this problem, there is no organized body knowledge about interoperability for AAL systems impacting decision making on which approach to use in developing this type of system.

A common solution to this scenario is using Non-Functional Requirements (NFR) catalogs. These catalogs enable the reuse of validated solutions while providing insight into their interdependencies and trade-offs [6]. In this work, we focus on creating a catalog of interoperability solutions, which summarize existing methods and technologies in the literature to mitigate this issue and identify suitable use cases. Interoperability is an example of a requirement with varying approaches depending on design decisions.

The need for interoperability arises due to the heterogeneity of devices, communication protocols, and programming languages used by different manufacturers, which should complement each other. In the literature, mature solutions to the interoperability problem exist, employing concepts and technologies of IoT, pervasive computing, and ubiquitous computing.

During the development of an AAL system, software engineers may encounter interoperability challenges at various levels (e.g., platform, device, semantic). Each situation requires its own set of solutions and design decisions that must align without being incompatible with one another. It's why we propose an interoperability catalog that characterizes the AAL landscape. Our catalog establishes correlations between the solutions offered for interoperability. This entailed cataloging strategies, subcharacteristics, architectural patterns, platform types, communication protocols, and semantic and syntactic aspects. The catalog was constructed by analyzing and extracting information from recent academic literature. In this paper, we utilize the following research question as a guideline:

[1] http://www.aal-europe.eu

R.Q.: ***What are the solutions in the literature that support interoperability in AAL systems?***

We followed the CORRELATE process [5] to create the catalog based on a set of 25 literature papers resulting from forward snowballing and systematic mapping studies. Data were extracted and coded using the Grounded Theory (GT) method [42] and were subsequently evaluated by two specialists.

At the end of the process, we developed the RECITAAL (REquirement Catalog of InteroperabiliTy Ambient Assisted Living systems), a Softgoal Interdependency Graphs (SIG) [6] for interoperability for AAL, comprising 70 subcharacteristics. The RECITAAL presents solutions found in the academic literature for each type of interoperability problem encountered in developing an AAL system. We evaluated this version of RECITAAL with a group of ten AAL specialists. They evaluated it in terms of clarity, readability, relevance and usefulness, as well as having space to suggest additions and improvements.

The remainder of the paper is divided as follows: Section 2 presents the definitions of interoperability for AAL and lists other requirements' catalogs. Section 3 describes the methodology used and its steps for building the catalog. In turn, Section 4 describes the RECITAAL, in addition to presenting the interrelationship table and an example of using the catalog. 5 presents the results of the evaluation with the 10 specialists in AAL and the Section 6 highlights the research contributions to AAL system developers and its threats to validity. Finally, Section 7 concludes the paper and details future work.

2 Background

2.1 Interoperability in AAL

In various contexts, we encounter the need for communication between systems, or even within the same system, built by different manufacturers or employing different standards and languages. The ability of these systems to establish connections and exchange data is referred to as interoperability. However, this data exchange process is often complicated due to multiple communication protocols and challenges related to semantics and syntax. According to IEEE, interoperability is "the ability of two or more systems or components to exchange information and to utilize the information that has been exchanged" [1].

The challenge of achieving interoperability is not a recent issue. The existence of heterogeneity in protocols, APIs, and platforms, along with the necessity of establishing communication among them, has been present since the early days of distributed computing [4].

While several solutions have been proposed and developed over the years, the evolution of computational systems and new paradigms, such as Pervasive Computing and the Internet of Things (IoT), has introduced an environment of extreme heterogeneity. However, attempts to establish standards to enable universal interoperability (e.g., OpenCCM and Web service standards) have faced significant challenges due to their high complexity [4].

From a macro perspective of healthcare systems, the development of strategies to address data interoperability problems in e-Health scenarios has become even more crucial with the increased integration between health systems [15] [17]. Such strategies are being encouraged by government initiatives in countries like Australia, Canada, the United Kingdom [17], and the United States [9].

Specifically within the context of the AAL domain, interoperability becomes a challenging issue due to the vast array of sensors and actuators employed in this scenario. In most cases, these devices come from different manufacturers and possess distinct internal architectures and communication protocols.

Solutions for interoperability in the AAL domain must take into account other requirements that are focused on the care and needs of older individuals, such as privacy, reliability, bi-directional communication, and mitigating false negatives. One commonly employed solution is the development of middleware platforms that centralize communication between sensors and other systems. Several works have explored this approach [14] [32] [44], often prioritizing different requirements [14].

2.2 Requirements Catalog

A requirements catalog is a valuable resource that organizes our past experiences, standard techniques, and knowledge of specific requirements. It captures the interdependencies, trade-offs, and associated concepts and terminology, enabling software engineers to navigate many development alternatives [6][5].

There are three types of requirements catalogs [6]: (i) Catalogs that represent knowledge about specific requirement types, non-functional requirements, or quality subcharacteristics, along with their associated concepts and terminology; (ii) Catalogs that capture implicit interdependencies (correlations, trade-offs) between softgoals. These are commonly referred to as correlation catalogs and (iii) Catalogs that systematically organize development techniques to assist software engineers in meeting specific requirements. These are often referred to as method catalogs.

Software engineers can access catalogs from the beginning of system development in typical scenarios. The catalogs can be extended or modified throughout the development process to accommodate new concepts or development techniques. Catalogs can take various forms of representation. SIGs provide a graphical method to structure the interdependencies of a given requirement [6].

Softgoals serve as the fundamental unit for representing requirements, non-functional requirements, and quality subcharacteristics. Their purpose is to assist software engineers in dealing with requirements that may be subjective, relative, and interactive in nature. Given that softgoals are interdependent, one method for addressing this challenge is the NFR Framework proposed by Chung. This framework represents softgoals and their interdependencies in SIGs, which serve as a graphical record of development decisions and design rationale [6].

3 Characterizing Interoperability for AAL

3.1 Methodology Overview

Figure 1 presents the methodology we followed to define interoperability in AAL systems. This methodology was based on the CORRELATE process [5]. This process supports the creation of requirements catalogs, from selecting the NFR to setting up the SIG. It consists of five steps: the first step starts with finding and selecting, within the academic literature, the papers that would serve as the basis for the catalog. To do so, we began with a systematic mapping of AAL systems, and from there, we performed a forward snowballing. From the selected papers, we extracted the data related to interoperability for AAL.

We applied the GT method to perform the qualitative analysis of the extracted data set [42]. Hence, the third step was open coding, where the data was summarized into codes and text segments. These codes were classified into four categories and evaluated by two experts in pervasive computing. After refining the codes, we performed the axial and selective coding steps. In this step, we correlated the codes and built the SIG.

3.2 Forward Snowballing

Our research started with a systematic mapping of AAL systems that were published in [10]. We found 35 papers that described AAL platforms using IoT and Cloud Computing solutions. We identified the essential needs of AAL, and its intended audience, and examined the most widely used architectural standards in AAL systems. To further investigate interoperability, we carried out a more focused exploratory study. We used a forward snowballing technique to identify studies that considered interoperability a requirement and update the studied list of papers. Integrating systematic mapping and the forward snowballing approaches form the basis for creating our interoperability catalog.

3.2.1 Protocol. To maintain standardization with the first stage (i.e., the systematic mapping), we decided to keep the same exclusion criteria of the study presented in [10]. Our only adaptation was to change the publication interval from 2018 to 2021. We aimed to reduce noise and eliminate the possibility of papers already being analyzed in the first stage reappearing.

We refined the inclusion criteria to suit the new scope better. The inclusion criteria for forward snowballing were as follows:

(1) The paper must present AAL systems that have interoperability as a requirement.
(2) The paper must address at least one of the following questions:
  (a) What are the interoperability definitions?
  (b) What necessitates the need for interoperability?
  (c) How is interoperability characterized?
  (d) How is interoperability implemented?

3.2.2 Execution. Figure 2 summarizes the results of each step of our snowballing process. Of the 35 papers selected in the final systematic mapping, 15 cited interoperability as an essential requirement for their AAL system development. These 15 papers served as the basis for our forward snowballing. Initially, we found 540 results that cited these papers. 162 papers were published after the analysis we did in the systematic mapping.

Out of these 162 papers, 30 papers presented AAL systems that required interoperability. Among these, 14 papers addressed at least one of the four questions related to interoperability. During the systematic mapping process, the initial 15 papers had already gone through the exclusion and inclusion criteria. These papers presented AAL systems that required interoperability. Therefore, in this set, we only analyzed which documents addressed in detail any of the four questions regarding interoperability. 11 papers passed this step.

Table 1: Example of softgoal

Code	Text segments from extracted data
Integration with major systems	*"Interoperability with a standard for sensor devices offers a great benefit for AAL systems, enabling the integration of a compatible sensor with little effort." [38]*
Different communication interfaces	*"Smart devices in smart homes usually come from different companies and do not use a single communication protocol to communicate their measurement and sensing data. This is the main reason of interoperability issue in smart home." [26]*
Flexible	*"The ability to embrace new sensors as well as actuators, and operate successfully is imperative." [2]*
Use of standards	*"However, to mandate the quality and ensure a consistency in the process of the development, standards gathering all the interesting aspects of the field are required to be available and considered." [2]*

By combining the two resulting sets, we obtained a total of 25 papers between 2013 and 2021 that described AAL systems in which interoperability played an important role.

The 11 papers selected from the mapping study were: [39] [43] [40] [18] [13] [20] [23] [7] [30] [8] [25]. 14 papers from the Snowballing were: [38] [11] [36] [19] [22] [24] [33] [26] [2] [3] [16] [37] [41] [34].

3.3 Qualitative Analysis and Open Coding

Following the GT method, the next step was the collection of data from the 25 papers [42]. One author selected excerpts from the 25 documents that answered one of the four questions. These text segments were initially classified into four large groups, one for each question. Thus, we acquired an initial set of 227 text segments concerning the causation, definition, characterization, and implementation of interoperability for AAL.

The next step was coding, which we divided into three parts: open, axial, and selective. In open coding, we started with text segments and extracted the core concepts of the codes. A code can contain a single word, an expression, or a phrase. These codes represent the concepts extracted from each text segment and, as a result, could be the same for several excerpts.

Table 1 illustrates an example of the code obtained for each question. To facilitate the encoding process, we employed the QDA Miner Lite tool. Its free version aided us in coding and analyzing the data extracted from the 25 papers.

Initially, we obtained 42 codes from 227 text segments. Then, two experts evaluated each of the codes and excerpts and classified them individually between "agree", "partially agree", "neutral", "partially disagree" or "disagree".

Those experts have Ph.D. and have researched pervasive computing for over a decade. Text segments rejected by both experts

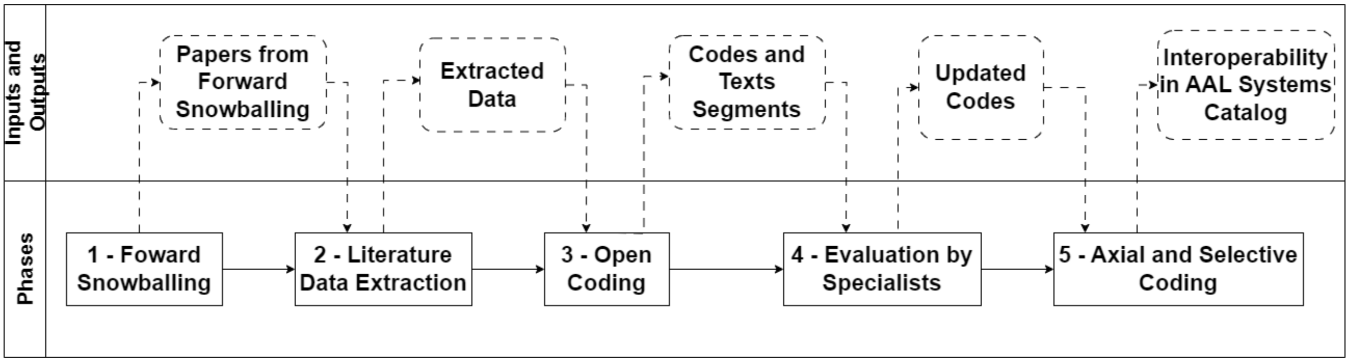

Figure 1: Methodology process

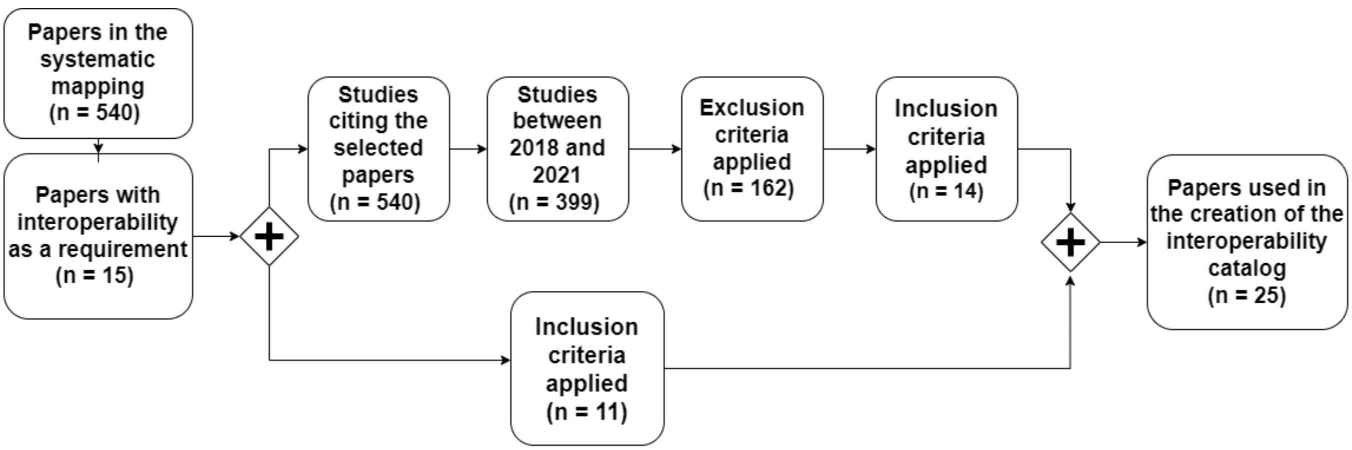

Figure 2: Forward Snowballing process

were excluded, and those approved by both were classified. If a code had all excerpts excluded, the code was excluded. A consensus meeting was held to analyze each of the cases in which there was disagreement between the two experts. At the end of this process, one code was excluded. The complete list of codes and text segments that were extracted from the papers and approved by both experts is at the following link: https://bit.ly/3MPkIRD

3.3.1 Axial and Selective Coding. The next step was axial coding. We identify codes that present similar concepts and group them into categories. Some codes from different questions were merged into one. In certain cases, categories were based on codes obtained in the previous step (ex: "Abstraction Layer" or "Message Protocol". In others, we create categories according to the grouped concepts. We also analyze the relationships between codes and categories using the contribution types provided in the SIG notation (AND, OR, BREAK, HELP, HURT, MAKE and UNKNOWN).

Figure 3 exemplifies the axial coding process. Categories "Protocol Network" and "Interoperability Standards" were created to group their respective codes. Then, the category "Communication" was designed to group the correlated categories. Thus, we constructed a knowledge pyramid comprising codes related to the solutions found in the literature for the various types of interoperability that could arise in AAL system communication.

So we created a chain of categories until we got to the core category, "Interoperability for AAL". This stage of the process is called Selective Coding, when all concepts (codes and categories) are united and refined. In the end, we only use the SIG notations AND ("—" over the line), OR (double "—" over the line), HELP ("+"), and MAKE ("++"). This process was performed by one researcher and evaluated by two specialists in pervasive computing and requirements (the same experts from the previous step).

Then, we started the process of refining the catalog, adding softgoals and operationalizations that were not found in the papers. Operationalizations are development techniques, operations, functions, data or constraints of a NFR softgoal, corresponding to the last levels of the catalogs [6]. This way, we can update the catalog with technologies that were not mapped in our coding. In total, in this version of the catalog, RECITAAL has 41 softgoals and 29 operationalizations.

We highlight the operationalizations with a stronger color. The complete list of subcharacteristics that are part of this catalog can be found in the following link: https://bit.ly/3R1XBpl

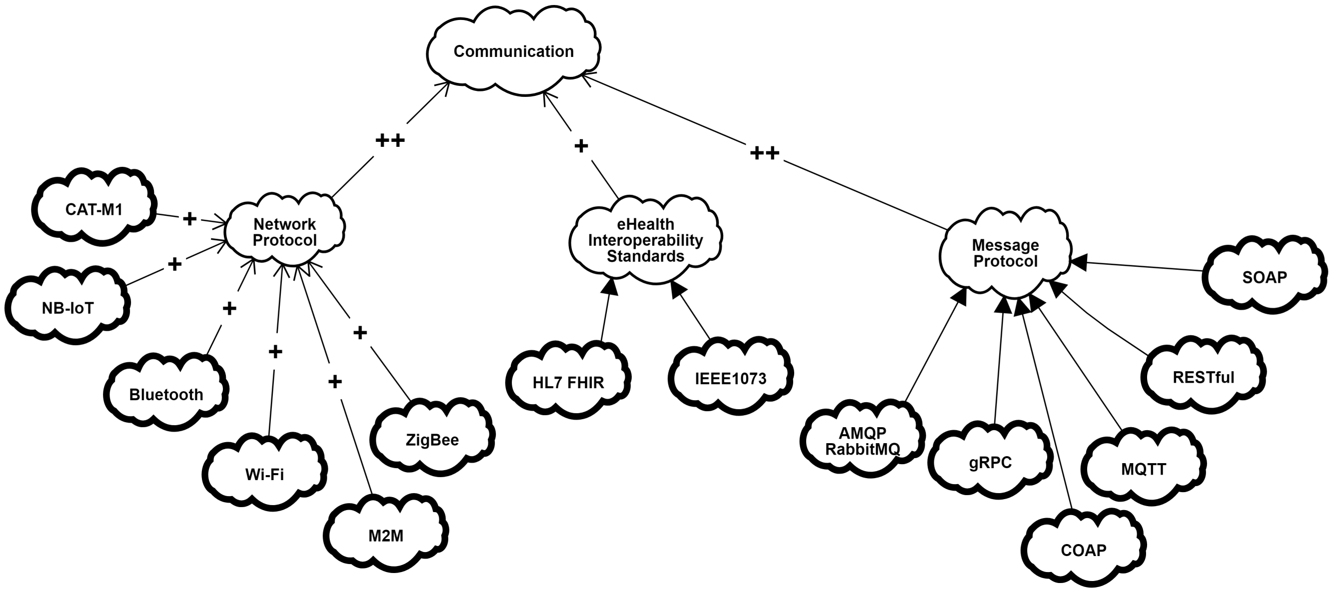

Figure 3: Example of axial coding grouping codes related to the Communication category

Table 2: Examples of codes from open coding

Framework	
Description	A standard structure that encapsulates a set of functions for the system.
Type	NFR softgoal
Positive interrelationships	Modular Architecture; Plug-in; Gateway; Wi-Fi; ZigBee; COAP; MQTT; Protocol Buffers; JSON; XML; Knowledge Base; Ontologies
Negative interrelationships	—

4 Catalog Interoperability for AAL

The RECITAAL, as depicted in Figure 4 [2], is focused on interoperability for AAL and was created to support software engineers who wish to develop AAL systems with interoperability as a key requirement. Your primary target audience is those who have no experience with interoperability.

The subcharacteristics of the RECITAAL cover various aspects and stages of design and requirements elicitation where interoperability issues may arise, along with commonly used solutions for each problem. Table 2 presents an example of a subcharacteristic, with its definition, type and what positive or negative interrelationships it may have.

4.1 Interoperability for AAL

In the context of the AAL domain, from this study, interoperability can be defined as *"the ability to communicate, collaborate and exchange data between different components of a single system or integrated systems"*. The primary challenges related to interoperability arise from heterogeneity, compatibility issues, and lack of standardization. These challenges can manifest at the macro level (integration between two systems) and micro level (semantic interpretation of exchanged messages). As a result, interoperability solutions can be categorized based on their type. This catalog has classified them into three broad categories: integration, communication, and data. Below are some examples of Catalog elements.

[2]This figure is also available at the link: https://bit.ly/4cHr0gL

4.2 Integration

The issue of interoperability arise from the challenge of integrating diverse systems or internally distinct components within the same system. The presence or absence of a hierarchy among systems (or system components) influence the mode of communication between them. We identified two **Types of Integration**:

- **Integration between different AAL systems:** Integration involves communication between two or more AAL systems.
- **Integration between components of a single AAL system:** Integration involves communication between the components of a single AAL system (e.g. sensors and the abstraction layer).

Such integrations must follow an **Hierarchy of Integration**, i.e. how communication between AAL systems (or the components of a single system) is done in architectural terms:

- **Vertical Integration:** The systems that must communicate are at different architectural levels.
- **Horizontal Integration:** The systems that must communicate are at the same architectural level.

Concerning the **Architecture**, we identified the architecture of the AAL system impacts the data exchange between its components

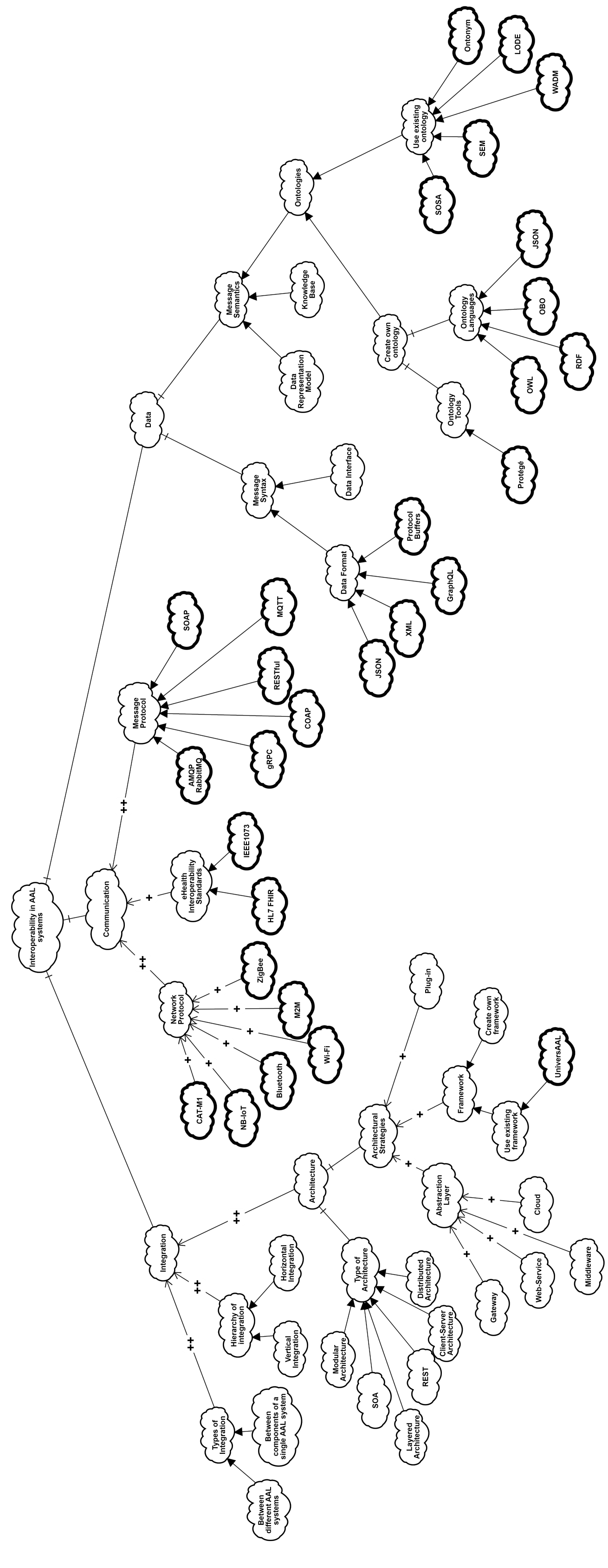

Figure 4: The Interoperability for AAL SIG

or its communication with external systems. Examples of type and strategies of architecture listed in the works:

- **Type of Architecture:** The type of architecture used by the system.
  - Client-Server Architecture; Distributed Architecture; Layered Architecture; Modular Architecture; REST; and SOA
- **Architectural Strategies:** The strategies used in the architecture to allow data exchange between parts of the system.
  - **Abstraction Layer:** Cloud; Gateway; Middleware; and Web Service.
  - **Framework:**
    - **Use existing framework:** UniversAAL[3]
    - **Create own framework**
  - **Plug-in**.

4.3 Communication

For two AAL systems, or two parts of a system, to exchange information, they need to be able to communicate with each other. Thus, the systems must follow certain communication standards. We identified three subcharacteristics of Communication: Network Protocol, Interoperability Standards, and Message Protocol.

For **Network Protocol**, AAL systems must adopt a common network protocol to allow communication between the subsystems (e.g, sensors, actuators). We identified three protocols listed in the papers (Bluetooth; Wi-Fi; and ZigBee[4]) and also added three other protocols (CAT-M1; M2M; NB-IoT) to the catalog.

Also, **e-Health Interoperability Standards** must be adopted. There are communication standards aimed at healthcare interoperability for example. We identified: HL7 FHIR[5]; and IEEE1073[6].

Additionally, AAL systems must be capable of interpreting the same **Message Protocol**, so the communication between the subsystems could be effective. Examples of message protocol solutions: AMQP RabbitMQ[7]; COAP[8]; gRPC[9]; MQTT[10]; RESTful[11]; and SOAP[12].

4.4 Data

Between two systems that need to share data, it is necessary that the exchange of messages is understandable for both sides. In this case, the **Message Syntax**, i.e., the structure of the message, must be interpretable by both systems. Message Syntax includes:

- **Data Format:** Use of data format standardized by the systems.
  - GraphQL[13]; JSON; Protocol Buffers[14]; and XML.
- **Data Interface:** Use of a standard for the body of the message (interface) that can be interpreted by both systems.

For the **Message Semantics**, i.e., the meaning of the message, should also be interpretable by both systems. For that, we identified as solutions:

- **Data Representation Model:** Use of data representation models, standardized among systems that exchange data, for mutual understanding of the meaning of each part of the message.
- **Knowledge Base:** Use of knowledge bases to interpret raw data.
- **Ontologies:** Use of ontologies for representation and inference of the exchanged data.
  - **Use existing ontology:** LODE[15]; Ontonym; SEM[16]; SOSA[17]; and WADM.
  - **Create own ontology**
    - **Ontology Languages:** OBO[18]; OWL[19]; and RDF[20].
    - **Ontology Tools:** Protegé[21].

4.5 Interrelationship Table

We also build a table of interrelationships between subcharacteristics. The purpose of this table was to map the interrelationships between subcharacteristics, that is, when the choice of one could affect the choice of another in a positive or negative form. To do so, initially, we returned to the 25 papers and mapped the subcharacteristics that tended to be used together in more than one study. Therefore, we classify these relationships as positive.

In the second part, we add the positive and negative interrelationships based on our experience with AAL and distributed systems in general. The third part consisted of requesting the AAL specialists who evaluated the catalog to add or remove interrelationships as they deemed appropriate. The current version of the interrelationship table is available at the link: https://bit.ly/3zKydyg

4.6 Usage Example

4.6.1 Motivating Scenario. Before we present an usage example of the RECITAAL, we need to define a motivating scenario and an application-example. The scenario is a hypothetical smart home with a living room, dining room, kitchen, bedroom and bathroom. This smart house had several sensors, actuators, and AAL systems, all with different manufacturers and communication protocols.

Six infrared presence sensors are distributed throughout the rooms. Developed in Java, each time they detect a movement, the sensors send a signal to the computer that controls the system. A Java application in the host computer needs to be responsible for decoding the message. The entire house uses smart lamps. The lamp control system has been developed in Python and is independent of the infrared sensor system. Three carpets are positioned in the living room and kitchen, with pressure sensors that communicate with the serve to a gateway encapsulated in a Raspberry Pi. This gateway also has a fall detection system developed in MatLab. The

[3]https://www.universaal.info/
[4]https://csa-iot.org/
[5]https://www.hl7.org/fhir/
[6]https://standards.ieee.org/ieee/1073/1568/
[7]https://www.rabbitmq.com/tutorials/amqp-concepts.html
[8]https://datatracker.ietf.org/doc/html/rfc7252
[9]https://grpc.io/
[10]https://mqtt.org/
[11]https://aws.amazon.com/en/what-is/restful-api/
[12]https://www.w3.org/TR/soap/
[13]https://graphql.org/
[14]https://protobuf.dev/
[15]https://github.com/essepuntato/LODE
[16]https://semanticweb.cs.vu.nl/2009/11/sem/
[17]https://www.w3.org/TR/vocab-ssn/
[18]https://obofoundry.org/
[19]https://www.w3.org/OWL/
[20]https://www.w3.org/2007/OWL/wiki/RDF-Based_Semantics
[21]https://protege.stanford.edu/

entire communication system with the pressure sensors was developed in C and is independent of the systems presented above. In this scenario, the main systems were developed using equipment from various manufacturers, each with their own communication protocols and programming languages.

4.6.2 Example of Use. Based on this motivating scenario, we designed an application-example with interoperability as the main requirement. This application-example should allow transparent interaction between the user and all sensors and actuators.

The first step is to define how communication should be between the sensors, actuators and the application. Given the relatively compact scale of the environment, consisting of a single house, we decided to use vertical integration, where the application has control of communication, and to adopt a layered architecture. To allow this communication between sensors/actuators and the application, an abstraction layer approach was chosen. In this context, Gateways serve as intermediaries responsible for transmitting and receiving data from sensors/actuators. And middleware functions as a layer that standardizes the data received by the application.

- **Integration:**
  - **Types of Integration:** Vertical Integration.
  - **Type of Architecture:** Layered Architecture.
  - **Architectural Strategies:** Abstraction Layer; Gateway; Middleware.

The next step involved determining the communication methodology. Given that home had a Raspberry Pi, to serve both as sensor gateways and as an application host, and our choice for a middleware, we decided to use AMQP RabbitMQ as message protocol. Due to the small scale of the project, we decided that, for this application-example, interoperability standards would not be used.

- **Communication:**
  - **Network Protocol:** Wi-Fi.
  - **Interoperability Standards:** No use.
  - **Message Protocol:** AMQP RabbitMQ.

Each sensor used its own unique patterns and vocabularies. Consequently, it is necessary to standardize syntax and semantics to understand the data. We choose to employ a data representation model. This model would encapsulate essential information from each sensor/actuator (e.g. entity identifier, timestamp, and raw data) into a unified data format within the application. In this case, Protocol Buffers was the selected data format.

- **Data:**
  - **Message Semantics:** Data Representation Model.
  - **Message Syntax:** Data Format; Protocol Buffers.

Thus, the RECITAAL comprehensively addresses all aspects of decision-making levels of the application-example. From the architectural choices to selection on the technologies and protocols used in communication and data exchange.

To support RECITAAL users, we created a template in the form of a questionnaire in which they fill out their decisions. The template is available at the link: https://bit.ly/3Y5l4Ki

5 Evaluation

To evaluate the catalog, we created a presentation about RECITAAL and an evaluation questionnaire. The presentation, in slides, presented the basic concepts about requirements catalogs, what the purpose of RECITAAL was and who its target audience was. At the end of the presentation, evaluators had access to the evaluation questionnaire link and the catalog artifacts (the SIG, the interrelationship table, the template and the table with the definitions of each subcharacteristic).

The evaluation questionnaire was based on the catalog evaluation by Moreira et. al.[28] and evaluated the catalog in terms of clarity, readability, relevance and usefulness. The questionnaire had a part with the demographic profile and self-assessment of the participants and a second part consisting of questions asked about the catalog.

There were four closed questions on a Likert Scale followed by five open questions where participants could suggest changes and improvements in each of the three main categories (integration, communication and data), for the interrelationship table and for the catalog as a whole.

The questionnaire was distributed using Google Forms, and participants were selected by convenience from personal contacts and lists of authors associated of related working documents about AAL. As our objective was for only specialists to evaluate, we eliminated participants who had less than one year of experience in the field. A total of 10 evaluators with more than one year of experience with AAL took part in the evaluation.

5.1 Profile of the Evaluators

The first part of the questionnaire was aimed at analyzing the profile of the evaluators. Thus, 80% of the evaluators had a doctorate, 10% had a master's degree and 10% were master's students. 80% of evaluators worked on developing at least one AAL system. Mainly in the areas of design, requirements and software development. Regarding experience with AAL, 20% of evaluators stated they had up to 3 years of experience. 30% said they had 3 to 5 years of experience in the area. 20% said they had 5 to 8 years of experience. Finally, 30% said they had more than 10 years of experience working with AAL.

5.2 Results

Figure 5 summarizes the results of Likert scale questions to participants. Adding up the positive responses (e.g., "agree" and "totally agree"), we obtained 90% approval for the criteria readability, relevance and usefulness. However, we received criticism about the clarity and usefulness of the catalog for the intended target audience. All evaluators made suggestions for improvement, mainly in the addition of new technologies in the Communication and Data categories. They made some suggestions, also considering the relationship between interoperability and other requirements, such as security. One of the evaluators, with more than ten years of experience in the area, also criticized the terms "positive" and "negative", believing that replacing them with more direct terms could better help catalog users to understand.

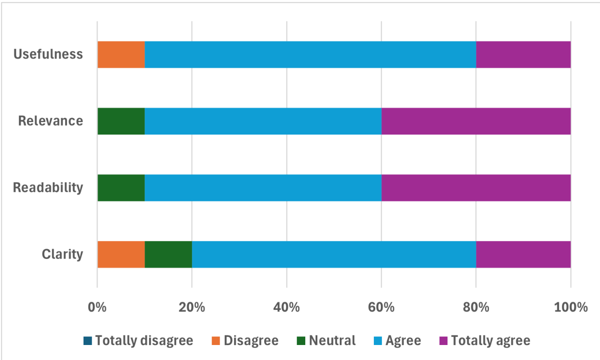

Figure 5: Catalog Evaluation

6 Discussions

6.1 Contributions

Through the CORRELATE process, we perform the RECITAAL, a SIG definition for interoperability in the AAL context. This RECITAAL acts to fill a gap in knowledge about interoperability for this domain. So we mapped interoperability types to this area. There are taxonomic studies on interoperability for IoT[31] and theoretical framework for context-aware software systems (CASS) [29], but there are no specific studies for the AAL domain.

In addition, it is important to use a more specific study for the AAL domain instead of approaches that analyze IoT more generally. A study analyzing the requirements of AAL systems concluded that the main requirement was the needs of the elderly population, also the main target audience of the systems [10]. The same study also observed that AAL systems did not use reference architectures for IoT. The analysis of subcharacteristics focused on healthcare, such as interoperability standards for healthcare, helps to differentiate between this catalog and one geared towards generic IoT systems.

The requirements catalog was produced to support software engineers interested in the development of AAL systems. The RECITAAL covers all the steps in which communication between parts of the system (or between different systems) can cause difficulties, pointing out the solutions commonly used in the literature for each situation.

6.2 Threats to validity

Although this research followed a specific process for creating a requirements catalog, there are threats to validity that must be considered. The first one concerns the group of papers considered in the research. Our research did not cover the entirety of the literature on AAL systems that have interoperability as a requirement. However, to mitigate this problem, we evaluate a set of papers, which started from a systematic mapping followed by a forward snowballing, which presents a portrait of literature between 2013 and 2021. Furthermore, the results have not yet been evaluated and/or refined by other experts in AAL systems. As the process of data extraction and coding using GT was performed by only one researcher, there is a possibility of bias. To mitigate this, we conduct an evaluation and refinement with pervasive computing and requirements experts during the coding stages. The objective is for the catalog to be constantly evolving. New technologies and approaches must be included, as well as technologies that have fallen into disuse must be removed. Therefore, although the base of the catalog consists of articles until 2021, the RECITAAL continued to be constantly updated, adding new items as feedback was received, both from authors and from evaluators and volunteers with experience in the area. Finally, another threat to validity is evaluation participants who had personal contacts with the authors. One way to mitigate this was to limit the evaluation to participants with a minimum amount of experience in AAL.

7 Final Considerations and Future Work

Interoperability is not a new issue in the context of software development in general. Indeed, several solutions and standards have been proposed over the last few decades. However, this caused another type of problem: there are several types of interoperability and not all solutions are compatible with each other. The RECITAAL serves as a catalog of requirements for the AAL domain. It maps the possible types of interoperability that can occur during the development of an AAL system and points out the most commonly used solutions for each situation.

The 25 papers that served as the basis for the construction of this catalog were obtained through a forward snowballing resulting from a systematic mapping, thus covering the academic literature on AAL systems between 2013 and 2021. Data extraction, coding, analysis of subcharacteristics and SIG assembly were carried out following the GT method and the CORRELATE process for producing requirements catalogs. Based on this core, the catalog was expanded with the addition of new softgoals and technologies that were not present in the papers.

In this way, through this SIG we answered the research question, *"What are the solutions in the literature that support interoperability in AAL systems?"*.

As future work, we plan to continue expanding and refining RECITAAL with additions from new sources beyond academic literature. The first step will be to make the catalog available on GitHub and encourage other researchers and developers to give their suggestions and improvements. The goal is to make RECITAAL more comprehensive for the types of interoperability possible during the development of AAL systems and to keep it up to date.

Acknowledgments

This research was partially funded by CNPQ, under grant number 314425/2021-7.

References

[1] 1991. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. *IEEE Std 610* (Jan 1991), 1–217. https://doi.org/10.1109/IEEESTD.1991.106963

[2] Abtoy Anouar, Touhafi Abdellah, Tahiri Abderahim, et al. 2020. A novel reference model for ambient assisted living systems' architectures. *Applied Computing and Informatics* (2020).

[3] David Bastos, José Ribeiro, Fernando Silva, Mário Rodrigues, Anabela G Silva, Alexandra Queirós, Antonio Fernández-Caballero, Nelson Pacheco Rocha, and António Pereira. 2020. SmartWalk BAN: using body area networks to encourage older adults to perform physical activity. *Electronics* 10, 1 (2020), 56.

[4] Gordon Blair and Paul Grace. 2012. Emergent middleware: Tackling the interoperability problem. *IEEE Internet Computing* 16, 1 (2012), 78–82.

[5] Rainara Maia Carvalho. 2019. *Correlate & lead: process and catalog of non-functional requirements correlations in ubicomp and iot systems*. Ph. D. Dissertation. Universidade Federal do Ceará.

[6] Lawrence Chung, Brian A Nixon, Eric Yu, and John Mylopoulos. 2012. *Non-functional requirements in software engineering*. Vol. 5. Springer Science & Business Media.

[7] Malcolm Clarke, Joost de Folter, Vivek Verma, and Hulya Gokalp. 2017. Interoperable End-to-End Remote Patient Monitoring Platform based on IEEE 11073 PHD and ZigBee Health Care Profile. *IEEE Transactions on Biomedical Engineering* (2017).

[8] Flavio Corradini, Francesco De Angelis, Barbara Re, Emiliano Anceschi, Massimo Callisto De Donato, and Paolo Iddas. 2017. Private assisted house for smart living. *Journal of Ambient Intelligence and Smart Environments* 9, 6 (2017), 725–741.

[9] John C Dodd. 2016. *Healthcare IT Transformation: Bridging Innovation, Integration, Interoperability, and Analytics*. Productivity Press.

[10] Paulo Duarte, E. F. Coutinho, J. Boudy, M. Hariz, and W. Viana. 2020. Using IoT in AAL Platforms for Older Adults: A Systematic Mapping. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. 705–710.

[11] Massimo Esposito, Aniello Minutolo, Rosario Megna, Manolo Forastiere, Mario Magliulo, and Giuseppe De Pietro. 2018. A smart mobile, self-configuring, context-aware architecture for personal health monitoring. *Engineering Applications of Artificial Intelligence* 67 (2018), 136–156.

[12] Laura Fiorini, Raffaele Esposito, Manuele Bonaccorsi, Claudio Petrazzuolo, Filippo Saponara, Roberta Giannantonio, Gianluca De Petris, Paolo Dario, and Filippo Cavallo. 2017. Enabling personalised medical support for chronic disease management through a hybrid robot-cloud approach. *Autonomous Robots* 41, 5 (2017), 1263–1276.

[13] Emanuele Frontoni, Rama Pollini, Paola Russo, Primo Zingaretti, and Graziano Cerri. 2017. HDOMO: Smart Sensor Integration for an Active and Independent Longevity of the Elderly. *Sensors* 17, 11 (2017), 2610.

[14] Nawras Georgi, Aline Corvol, and Régine Le Bouquin Jeannès. 2018. Middleware Architecture for Health Sensors Interoperability. *IEEE Access* 6 (2018), 26283–26291.

[15] Evelyn Johanna Sophia Hovenga. 2008. Importance of achieving semantic interoperability for national health information systems. *Texto & Contexto-Enfermagem* 17, 1 (2008), 158–167.

[16] Abdul Jaleel, Tayyeb Mahmood, Muhammad Awais Hassan, Gulshan Bano, and Syed Khaldoon Khurshid. 2020. Towards medical data interoperability through collaboration of healthcare devices. *IEEE Access* 8 (2020), 132302–132319.

[17] Craig E Kuziemsky. 2015. A multi-tiered perspective on healthcare interoperability. In *Standards and Standardization: Concepts, Methodologies, Tools, and Applications*. IGI Global, 1166–1181.

[18] Sofoklis Kyriazakos, Ramjee Prasad, Albena Mihovska, Aristodemos Pnevmatikakis, Harm op den Akker, Hermie Hermens, Paolo Barone, Alessandro Mamelli, Samuele de Domenico, Matthias Pocs, et al. 2017. eWALL: An Open-Source Cloud-Based eHealth Platform for Creating Home Caring Environments for Older Adults Living with Chronic Diseases or Frailty. *Wireless Personal Communications* 97, 2 (2017), 1835–1875.

[19] Wei Li, Cheolwoo Jung, and Jongtae Park. 2018. IoT healthcare communication system for IEEE 11073 PHD and IHE PCD-01 integration using CoAP. *KSII Transactions on Internet and Information Systems (TIIS)* 12, 4 (2018), 1396–1414.

[20] Christophe Lohr and Jérôme Kerdreux. 2020. Improvements of the xAAL home automation system. *Future internet* 12, 6 (2020), 104.

[21] Diego López-de Ipiña, Xabier Laiseca, Ander Barbier, Unai Aguilera, Aitor Almeida, Pablo Orduña, and Juan Ignacio Vazquez. 2009. Infrastructural Support for Ambient Assisted Living. In *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*, Juan M. Corchado, Dante I. Tapia, and José Bravo (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 66–75.

[22] Michael H McGillion, Emmanuelle Duceppe, Katherine Allan, Maura Marcucci, Stephen Yang, Ana P Johnson, Sara Ross-Howe, Elizabeth Peter, Ted Scott, Carley Ouellette, et al. 2018. Postoperative remote automated monitoring: need for and state of the science. *Canadian Journal of Cardiology* 34, 7 (2018), 850–862.

[23] Vittorio Miori and Dario Russo. 2017. Improving life quality for the elderly through the Social Internet of Things (SIoT). In *Global Internet of Things Summit (GIoTS), 2017*. IEEE, 1–6.

[24] Vittorio Miori, Dario Russo, and Luca Ferrucci. 2018. Supporting active aging through a home automation infrastructure for social internet of things. *Advances in Science, Technology and Engineering Systems Journal* 3, 4 (2018), 173–186.

[25] Gianfranco E Modoni, Mario Veniero, Alberto Trombetta, Marco Sacco, and Susanna Clemente. 2018. Semantic based events signaling for AAL systems. *Journal of Ambient Intelligence and Humanized Computing* 9, 5 (2018), 1311–1325.

[26] Ghassem Mokhtari, Amjad Anvari-Moghaddam, and Qing Zhang. 2019. A new layered architecture for future big data-driven smart homes. *Ieee Access* 7 (2019), 19002–19012.

[27] D. Monekosso, F. Florez-Revuelta, and P. Remagnino. 2015. Ambient Assisted Living [Guest editors' introduction]. *IEEE Intelligent Systems* 30, 4 (July 2015), 2–6. https://doi.org/10.1109/MIS.2015.63

[28] Ana Moreira, João Araújo, Catarina Gralha, Miguel Goulão, Isabel Sofia Brito, and Diogo Albuquerque. 2023. A social and technical sustainability requirements catalogue. *Data & Knowledge Engineering* 143 (2023), 102107.

[29] Rebeca C Motta, Káthia M de Oliveira, and Guilherme H Travassos. 2019. A conceptual perspective on interoperability in context-aware software systems. *Information and Software Technology* 114 (2019), 231–257.

[30] Sao Mai Nguyen, Christophe Lohr, Philippe Tanguy, and Yiqiao Chen. 2017. Plug and play your robot into your smart home: Illustration of a new framework. *KI-Künstliche Intelligenz* 31, 3 (2017), 283–289.

[31] Mahda Noura, Mohammed Atiquzzaman, and Martin Gaedke. 2019. Interoperability in internet of things: Taxonomies and open challenges. *Mobile networks and applications* 24 (2019), 796–809.

[32] Eduardo Osório, Liliana Ferreira, Rui Abreu, and Filipe Sousa. 2013. Interoperability in ambient assisted living using openehr. In *2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013)*. IEEE, 394–398.

[33] Panagiotis Papadakis, Christophe Lohr, Marin Lujak, Abir Karami, Ioannis Kanellos, Guillaume Lozenguez, and Anthony Fleury. 2018. System design for coordinated multi-robot assistance deployment in smart spaces. In *2018 Second IEEE International Conference on Robotic Computing (IRC)*. IEEE, 324–329.

[34] Nidhi Pathak, Sudip Misra, Anandarup Mukherjee, and Neeraj Kumar. 2021. HeDI: Healthcare device interoperability for IoT-based e-health platforms. *IEEE Internet of Things Journal* 8, 23 (2021), 16845–16852.

[35] P. Rashidi and A. Mihailidis. 2013. A Survey on Ambient-Assisted Living Tools for Older Adults. *IEEE Journal of Biomedical and Health Informatics* 17, 3 (May 2013), 579–590. https://doi.org/10.1109/JBHI.2012.2234129

[36] Danilo FS Santos, Mateus Assis M Lima, Jose L Nascimento, Angelo Perkusich, and Hyggo O Almeida. 2018. Enabling the first step for IoT health systems using Antidote and IEEE 11073. In *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE, 308–313.

[37] Fahad Ahmed Satti, Taqdir Ali, Jamil Hussain, Wajahat Ali Khan, Asad Masood Khattak, and Sungyoung Lee. 2020. Ubiquitous Health Profile (UHPr): a big data curation platform for supporting health data interoperability. *Computing* 102, 11 (2020), 2409–2444.

[38] Michael Schmidt and Roman Obermaisser. 2018. Adaptive and technology-independent architecture for fault-tolerant distributed AAL solutions. *Computers in biology and medicine* 95 (2018), 236–247.

[39] Jatinder Singh and Jean M Bacon. 2014. On middleware for emerging health services. *Journal of Internet Services and Applications* 5, 1 (2014), 6.

[40] Thanos G Stavropoulos, Georgios Meditskos, and Ioannis Kompatsiaris. 2017. Demaware2: Integrating sensors, multimedia and semantic analysis for the ambient care of dementia. *Pervasive and Mobile Computing* 34 (2017), 126–145.

[41] Thanos G Stavropoulos, Georgios Meditskos, Ioulietta Lazarou, Lampros Mpaltadoros, Sotirios Papagiannopoulos, Magda Tsolaki, and Ioannis Kompatsiaris. 2021. Detection of Health-Related Events and Behaviours from Wearable Sensor Lifestyle Data Using Symbolic Intelligence: A Proof-of-Concept Application in the Care of Multiple Sclerosis. *Sensors* 21, 18 (2021), 6230.

[42] Anselm Strauss and Juliet Corbin. 1998. Basics of qualitative research techniques. (1998).

[43] Arezoo Tahmasbi, Sahar Adabi, and Ali Rezaee. 2016. Behavioral Reference Model for Pervasive Healthcare Systems. *Journal of medical systems* 40, 12 (2016), 270.

[44] Diana C Yacchirema, Carlos E Palau, and Manuel Esteve. 2017. Enable IoT interoperability in ambient assisted living: Active and healthy aging scenarios. In *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 53–58.