

# RC-ASEF: An open-source tool-supported requirements elicitation framework for context-aware systems development

Unai Alegre-Ibarra, Juan Carlos Augusto, Carl Evans  
Middlesex University, London NW4 4BT, U.K.  
Email: {U.Alegre, J.Augusto, C.Evans}@mdx.ac.uk

**Abstract**—In terms of software engineering, context-aware systems (C-AS) have notably different development needs than those of traditional computing. Yet, there are no established methodologies that uniformly support the development life-cycle of these systems. A key goal of this research is to improve the current state-of-the-art with respect to engineering techniques for the life-cycle of a C-AS. Within the scope of this higher order goal, this paper addresses the lower level order goal of a holistic framework for gathering requirements which is specialised to the creation of C-AS. The framework follows an end-user, stakeholder-centred vision, which guides the analysis of stakeholders towards the discovery of specific stakeholder profiles and their particular needs, preferences, and limitations. It allows the operationalisation of the high level objectives of the system into requirements, which are more tangible and related to the implementation of the system. An evaluation procedure is supported, based on heuristics and rules from the NFR framework and REUBI. All the diagrams introduced for this framework have been developed as part of an open-source tool based on Modelio, which is intended to be developed in the future as part of a framework that covers all the stages of the development process. The proposal is illustrated through the analysis of an application for a European funded project.

## I. INTRODUCTION

The creation of context-aware systems (C-AS) can entail a great amount of challenge and complexity [1]. In comparison to the development of traditional systems, C-AS are more expensive, diverse, and prone to change. During its creation, developers might find difficult, or even impossible to identify the situations in which to display services, as well as what services should be exhibited in those situations [2]. Also, developers might find it challenging to describe the information to identify these situations, and make the system aware of them by using a heterogeneous array of sensors, which are likely to provide inaccurate, overlapping, contradictory or missing data. Additionally, advanced reasoning techniques need to be implemented in order to make the system infer that situations are happening, based on sensor information. This intricacy emphasises that there is a substantial difference between developing conventional systems and those that are context-aware. As part of previous research, an extensive analysis has studied the different approaches to the development of C-AS [1]. Although there is lot of research related to the development of these kinds of systems, this is focused on solving particular issues and it is usually scattered and not connected with other

development stages. The evidence presented in [1] supports the need of a more holistic and unified approach for the development of C-AS.

A key goal of this research is to improve the current state-of-the-art with regard to techniques and methods to help establish the foundations of a uniform engineering process that covers the entire life-cycle of a C-AS. As part of a lower-order goal of the bigger picture, this paper focuses on the creation of the foundations for the *Requirements for Context-Aware Systems Framework* (RC-ASEF), a holistic framework for the requirements elicitation stage, which takes into account the specific demands of C-AS development. The support provided for the requirements elicitation stage in RC-ASEF is divided into two main foci. During early stages of the requirements elicitation process, the methodology is focused on the generic or non-contextual aspects of the system ( $F_1$ ), to then iteratively advance towards the requirements which are more related to the identification of situations (context), the way in which they are planned to be detected by the system, and their associated context-aware features ( $F_2$ ). Previous research towards the high order goal of this work has focused on  $F_2$ , creating a deeper analysis into the conceptualisation of context and context-awareness [2], which takes into account the philosophical limitations of C-AS in order to create a perspective for developing more usable C-AS. The aim of the work presented in this paper is focused on a generic methodology for gathering the non-contextual aspects of a C-AS, corresponding to  $F_1$ , reusing existing methods and tools to provide a coherent requirements elicitation methodology that can cover the demands of C-AS development. Particularly it has been developed with reference to previous work [3] [4] [5], including the mentioned conceptualisation of context and context-awareness [2]. In particular, the framework is based on a collection of models, presented as a combination of dynamic and static diagrams which collectively define this new requirements elicitation framework, for which in addition, new, open-source tools have been developed. These constructs have been strategically chosen to be based on UML profiles, as this facilitates its use along with other existing standards such as UML [6], SySML [7], and U2TP [8], or other UML-based requirements profiles such as UML-AT [9]. These diagrams have been developed as an extension of the open-source tool

framework RC-ASE, Modelio<sup>1</sup>, which is further introduced in [2], and which can be found in [10]. The remainder of the paper is as follows. Section II analyses previous work in relation to the framework. Section III introduces the requirements elicitation framework. Section IV is related to the establishment of a project scope. Section V corresponds to the stakeholder analysis of the methodology. Section VI is related to the objective establishment activity. Section VII corresponds to the identification of functional requirements. Section VIII corresponds to the evaluation activities of the methodology. Finally, Section X concludes the paper.

## II. RELATED WORK

Previous work related to requirements which are specialised for C-AS can be divided into three main groups: scenario-based, goal-oriented and hybrid approaches. Related work has been reviewed [1] in order to find an approach that can be suitable for creating a more holistic approach, which reuses the most positive aspects of existing methodologies and tools. This paper focuses on an analysis of methodologies for a number of aspects that are considered relevant for the higher order goal of creating a more holistic framework for C-AS development, as shown in Table I. Columns 4, 5 and 6 focus on the coverage of the methodologies for the most common elicitation activities for C-AS development. These activities are covered by most of the analysed methodologies. Columns 8 and 10 represent whether or not the methodology is based on goals, scenarios or a hybrid approach. Column 9 describes if the methodology has support for the partial satisfaction of goals rather than just being binary. Column 11 indicates whether or not the methods provide specific and systematic treatment of non-functional requirements. The most complete approach is that of REUBI [3], which has the potential to cover all these approaches. Column 12 indicates whether or not the methodology takes into account the needs, preferences and limitations of the end-users, or in its absence, they support personalisation to a certain degree. Only three methodologies support this feature, from which PC-RE [4] and R4IE [5] are highlighted. Column 13 shows whether or not the methodologies take into account the influence of contextual aspects. Many methodologies support this, but each has its own particular way to manage this. Column 14 indicates if the methodology has specific support guiding developers into: (a) enumerating the set of contextual states that may exist, (b) knowing what information could accurately determine a contextual state within that set, and (c) stating what appropriate action should be taken from a particular state [2]. Oyama et al. [11] present a series of templates for this purpose which could be reused for other methodologies. Columns 15 and 16 show whether or not the approaches have tool support and if such tool is freely and easily available for other researchers to be extended.

From the point of view of the analysis of those aspects, REUBI [3] is the most complete methodology, as it can

be observed in Table I. Nevertheless, there are three main aspects that this method does not completely cover. Namely, the explicit lack of a user-centred perspective, and a lack of a tool which is publicly available. Also, it does not provide guidance for developers to discover context, according to the three main principles to get the context right [12] [2]. From the point of view of guiding the developers towards the discovery of situations and context [2], the data-information-knowledge-wisdom model of Oyama et al. [11] could also be employed for this purpose. Nevertheless, Oyama's model lacks mechanisms for elaborating and modelling requirements. Compared to REUBI [3], it also lacks mechanisms for handling soft goals and non-functional requirements. Additionally, there is no tool support for this approach. For the purpose of this research, the REUBI methodology [3] is the most relevant reference point. Therefore, it is concluded that REUBI will be used as the foundation from which the requirements framework for engineering C-AS will be built. Although REUBI has partial support for scenario based techniques, which can be used to understand and gather the context of the system, scenario-based techniques are not a necessary requirement for this work, and they can be further complemented with other techniques. A necessary aspect that needs to be covered for this work, is that of the user-centred perspective. This gap can be addressed by combining other existing methodologies. The R4IE [5] and PC-RE [4] approaches have some synergies that can be used to complement this characteristic. Additional techniques for analysing stakeholders and their needs can also be useful for this purpose. In order to address the shortcomings related to guiding developers into context discovery, a set of guidelines which are based on the perspectives of [2] will be included as part of the methodology. Finally, and significantly, whilst the REUBI approach has no explicit open-source tool support, the work described here has a specific goal of developing an open-source tool to support the proposed software development framework, which includes specific support for requirements engineering.

## III. RC-ASEF: REQUIREMENTS FOR CONTEXT-AWARE SYSTEMS ENGINEERING FRAMEWORK

Figure 1 presents the six main activities of a coherent methodology out of the most relevant approaches identified for the purpose of creating a framework for supporting the non-contextual aspects of the requirements elicitation, influenced by R4IE [5]. The main enhancement is that the identification of system performance qualities, used for gathering non-functional requirements, is now part of the objective establishment. A new activity group, corresponding to the evaluation of the objectives and requirements, which is partially based on the harmonisation activity from R4IE, is introduced. The activities in Figure 1 are divided into different sub-activities, as shown in Figure 1, which are mainly influenced by the works presented in [3] [5]. The method gives great importance to the exhaustive analysis of the stakeholders of the systems, as part of the identification of their needs and preferences in further stages. The sub-activities constitute an enhancement

<sup>1</sup><https://www.modelio.org/>

No.	Name	Year	Reference	Acquisition	Elaboration	Modelling	Goal-oriented	Softgoals	Scenario based	NFR treatment	User-centred	Context-awareness	Developer Guidance for context	Tool support	Open source
(a)	(1)	(2)	(3)	(4)	(5)	(6)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)
(a)	-	2001	[13]	-	✓	✓	✓	-	-	-	-	✓	-	-	-
(b)	PC-RE	2006	[4]	✓	✓	✓	✓	-	✓	-	✓	✓	~	-	-
(c)	RE-CAWAR	2007	[14]	-	✓	✓	✓	-	✓	-	-	✓	~	-	-
(d)	-	2008	[11]	✓	-	-	✓	-	-	-	✓	✓	✓	-	-
(e)	-	2008	[15]	✓	-	~	✓	-	✓	-	~	✓	~	✓	-
(f)	FLAGS	2010	[16]	-	✓	✓	✓	✓	-	-	-	-	~	✓	-
(g)	REUBI	2013	[3]	~	✓	✓	✓	✓	~	✓	~	✓	~	✓	-
(h)	R4IE	2014	[5]	✓	✓	-	~	-	~	~	✓	✓	-	-	-

✓ = The property is completely satisfied. ~ = The property is partially satisfied. "-" = The property is not satisfied at all.

TABLE I

COMPARISON OF CURRENT METHODOLOGIES FOR REQUIREMENTS ENGINEERING IN TRADITIONAL AND CONTEXT-AWARE SYSTEMS.

of the R4IE methodology, where the first sub-activity of the stakeholder analysis is inspired by [17], and the second sub-activity is impacted by the profiling of users [4] [5], the ethical analysis recommendation in [17], and the e-FRIEND ethical framework [18]. It is also influenced by the conceptualisation presented in [2]. The last activity in the stakeholder analysis, and the sub-activities related to the establishment of objectives, have been adopted from [3]. Finally, those sub-activities corresponding to the identification of functional requirements and the application of the evaluation procedure are inspired by those activities in [3], and influenced by the heuristics and rules from the NFR Framework [19] as well as the SySML [7] standard.

#### IV. ESTABLISH SCOPE

The central activity of the methodology during  $F_1$ , is to establish the scope of the system in terms of the system boundaries (*i.e.*, what is inside the system and what is immediately external to it). As it can be observed in [5], this activity is influenced by the remaining core activities in  $F_1$ , which help to determine the objectives, resources, budget and schedule to be included within the scope statement.

#### V. STAKEHOLDERS ANALYSIS

The initial step consists of a stakeholder analysis, which allows documenting and modelling the outcome from the array of techniques proposed in [17], using a UML profile for the creation of Stakeholder Diagrams. The stakeholders are identified, and their different relevant relationships to the project are analysed. The outcome of this activity is used as part of the scope statement and part of the models. Finally, the aim is to identify different user profiles, in order to pave the way for discovering useful Situations of Interest in  $F_2$ . Using the information gathered during the stakeholder analysis, it focuses on the identification of activities.

##### A. Identify stakeholders

This activity is initiated by a small group, and later reviewed with a larger group of stakeholders. After the review with

a larger group of stakeholders, the participants should think about those stakeholders who are still not included. If there are more interested parties, a bigger group should be assembled to review the stakeholders [17]. This process iterates until a consensus has been arrived at such that it is considered that all relevant stakeholders have been accounted for. A set of techniques are recommended to guide this process [17]. Each of which can build on the previous technique, and it includes the listing of stakeholders, its basic analysis, the power versus interest grids as well as the stakeholder influence diagrams. The stakeholder identification task can also be complemented with a stakeholder analysis, as further explained in [17].

##### B. Determine stakeholder profiles

The aim of this activity is to identify stakeholder profiles, by establishing personal goals and setting different levels of achievement. The user profiling is attained by setting certain achievement levels and monitoring progress towards those personal goals [4]. In order to set the achievement levels, three main dimensions are analysed during  $F_1$ , which include the cultural aspects of the stakeholders, their quotidian activity, and their relevant ethical aspects. Finally, the information obtained from this analysis is used to customise the requirements, as well as the system set-up and training. In activities related to stakeholder profiling corresponding to  $F_2$ , other dimensions are analysed, namely, the interaction modalities, and the mechanisms for monitoring the achievement of personal user goals. The user profiling activity is mainly based on the activity with the same name in R4IE [5], but it also includes the cultural analysis and profiling guidelines from PC-RE [4] and the ethical analysis mechanisms from [17] and [18]. The main enhancement is that the *task subset* and *context-interaction requirements* sub-activities of R4IE [5], and the *monitoring mechanism specification* related activities of PC-RE [4] have been moved to the context-aware specialised stage,  $F_2$ . Also, a new sub-activity has been proposed, to analyse the activity of stakeholders in order to prepare the situation of interest identification in  $F_2$ .

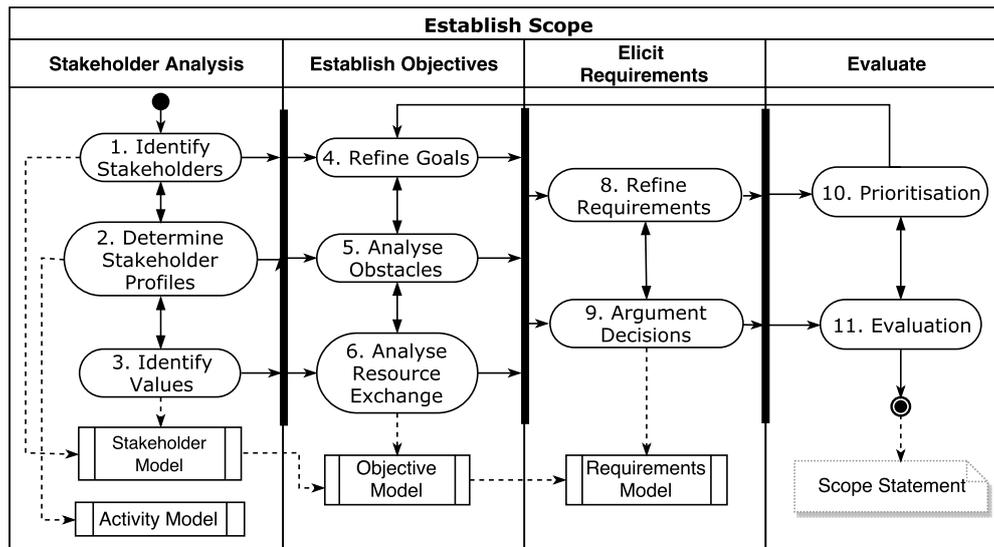


Fig. 1. Activity model representing the core activities and corresponding sub-activities in the early requirements elicitation stage, corresponding to F1.

1) *Cultural analysis*: The first sub-activity of stakeholder profiling deals with the system from an international point of view, where the different effects of culture are analysed in order to influence the definition of requirements for localising systems and specifying how it will be tailored for its different cultural profiles. During this activity, scenarios are sourced from users who belong to the cultures, nationalities and linguistic groups inside the intended market. The four<sup>2</sup> main steps of the guide proposed in [4] can be applied for this purpose.

2) *Ethical analysis*: An ethical analysis can contribute to ensure the ethical appropriateness of actions are ultimately taken in a project. For this purpose the use of Ethical Analysis Grids is recommended [17]. This grid can aid the satisfaction of both deontological (duty-based) and teleological (results-oriented) obligations. It consists of classifying some characteristics of each stakeholder into: High, Medium, Low and None. The characteristics are the vulnerability and gravity of the stakeholder, her/his dependency on the government, likelihood remedy, risk to fundamental value and policy impact. Although the ethical analysis proposed in [17] is useful for general purpose systems, it is not focused on C-AS. Context-awareness is the essence of different areas<sup>3</sup> that typically raise some ethical concerns which are different to those of traditional systems. For this reason, this sub-activity also adopts the eFRIEND ethical framework [18]. In order to apply it, it is recommended to carefully evaluate and discuss with the end-user stakeholders the different ethical concerns that might arise, until there is an agreement between all parties (e.g., increasing user safety at the expense of giving up some privacy). The discussions can be complemented by questionnaires or

interviews. The outcome of those discussions at a conceptual level can be used to modify or create different objectives and requirements.

3) *Activity analysis*: This stage consists of analysing the activity of end-user stakeholders, and is especially focused on that activity of end-user stakeholders. The purpose is to facilitate (for the benefit of developers) the identification of the meaning behind the behaviour of the end-user stakeholders. Particularly, by analysing how they usually behave in their quotidian tasks, and by thinking about how the stakeholders could use the proposed system to improve the way in which they achieve these tasks. This gives more opportunities to identify services that can be provided to them according to their particular needs, preferences, and limitations. Techniques such as observation, prototyping, scenarios or *wizard of oz* [20] can be used. Other approaches such as ethnomethodology can be adopted to understand the meaning of the actions of the end-user stakeholders. On the other hand, data analysis techniques such as classification or pattern-recognition could also help in revealing unexpected relations in the behaviour of the stakeholders.

4) *Determine customisation, set-up, and training*: The method proposed in Figure 1 is iterative. Once developers have defined some requirements, it is time to use the information gathered during this activity to customise existing requirements. The main dilemma is to specify C-AS that suit the requirements of individual users, while delivering a general system that can be used by many (individually different) users [4]. Not only this, but requirements can also evolve for the same user. For instance, as users become more experienced using the system, they require less help and supportive dialogues, and can access more sophisticated features. Also, the requirements engineering process should take into account aspects of maintenance and bespoke tailoring (to different stakeholders) after the system is deployed [5]. In

<sup>2</sup>Note that the fifth step has been moved to the user profiling activity in F<sub>2</sub>.

<sup>3</sup>Particularly referring, in this case, to Ubiquitous & Pervasive Computing, Intelligent Environments, Ambient Intelligence and Ambient Assisted Living.

order to help the identification of different stakeholder profiles, there is a need to think about how the system will be set-up by/for the different stakeholders, trying to distinguish the different common needs of stakeholders that can be classified into profiles. As well as how the different stakeholders will want to customise the system, what type of training will they receive, and how will they receive it. In order to enable the customisation of the system, an individual user profile is defined first. The requirements are frequently set by another expert stakeholder (e.g., a teacher sets certain requirements for a student's learning abilities). Individuals directly elicit (and own) personal goals. For both personal goals and user profiles, attainment targets can be set which become benchmarks for monitoring processes. A trade-off analysis might help to identify any conflicting user profile goals set by the expert stakeholders with the personal ambitions of the end-user stakeholder.

### C. Identify values

During this sub-activity the aim is to identify the different values to be produced by the system and consumed by the stakeholders. This model has been adopted from the value model introduced in REUBI [3]. It takes existing stakeholders, humans or agents, and identifies the different values that are expected to be exchanged. Particularly, stakeholders which produce, consume a value, or are interested in a value or in its acquisition quality. Also, developers analyse what values are interchanged by the system and other stakeholders. Then developers reflect on how these values can be enhanced. Aspects such as how to improve the value, what is the expected quality of the value, what time restrictions exist in the provision of the value are taken into account. Other enhancement aspects apply, such as the flexibility in the value acquisition, precision or reliability restrictions, as well as cost or security restrictions applicable to the value.

### D. Stakeholder diagram

Inspired by the techniques of the sub-activities explained in this section, the Stakeholder Diagram is introduced, which can model relevant stakeholder related information, as it is shown in the meta-model of Figure 2. In addition to the meta-models of this diagrams, each stakeholder can be associated to a *UserProfile* stereotyped element. This user profile is composed of *profileFeature* stereotyped elements that can have *profileFeatureInstances*. This is better illustrated in the example from Section IX-A.

## VI. ESTABLISH OBJECTIVES

Taking into account the value analysis, the objectives of the system are declared. Then, from the higher order objectives, a refinement process is applied in order to obtain and decompose them into sub-objectives. This step is followed by an analysis of the adverse conditions that may impede the satisfaction of a goal. Following this, the analysis focuses on the resources required for the satisfaction of goals.

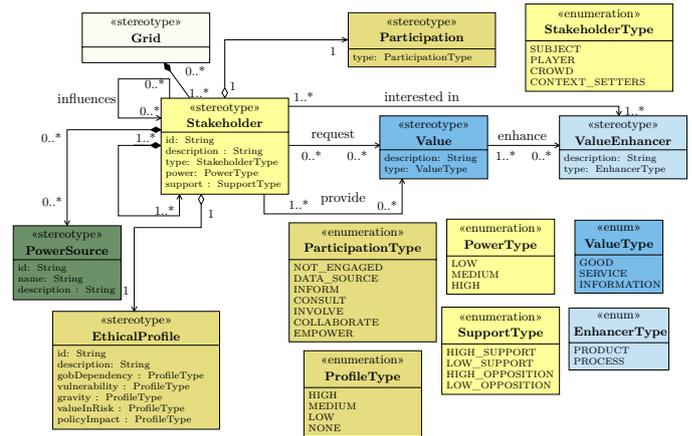


Fig. 2. UML meta-model for the Stakeholder Diagram.

### A. Refine goals

Once the system boundaries and the higher-order objectives are identified, in the form of *values* and *value enhancers*, it is necessary to derive more specific objectives, and progressively refine them in order to obtain more knowledge about the system under development. Objectives act as a bridge between the system values and the final requirements of the system, providing specific guidance during the requirements elicitation process. Objectives are divided into *goals* and *soft-goals*, according to the identified *values* and *value enhancers*. Goals have a clear criteria of satisfaction. Soft-goals do not have a clear criteria of satisfaction, which means that they can be used for identifying and modelling non-functional requirements. The objectives can also be progressively decomposed using *inclusive* (AND), *alternative* (OR), or *exclusive* (XOR) relationships between them. Such relationships can help in determining if their corresponding parent objectives are satisfied or not, as further explained in [3].

### B. Analyse obstacles

The dynamic nature of C-AS is closely related to the existence of multiple adverse conditions which can make it difficult for system objectives to be met. This sub-activity consists of identifying obstacles which may affect meeting a specific goal, in the same way as described in [3]. The main objective is to determine those situations which are likely to be inconvenient for meeting the objectives, even if obtaining a complete set of adverse conditions can be a difficult achievement.

### C. Analyse resource exchange

Sometimes, there exist restrictions on the way in which sub-objectives need to be satisfied in order to satisfy the parent objective; such as not satisfying an objective until other objectives are satisfied, mainly because these require access to certain resources which are generated as a result of satisfying other objectives. The objectives relate to the resources through two different relationships: *provision* and *demand*, as further explained in [3].

#### D. Objective diagram

With the purpose of facilitating the objective, obstacle and resource exchange analysis explained in this section, the *Objective Diagram* is introduced, which has been adopted from the Interdependency Graph in [3]. The meta-model of this diagram and its corresponding example can be observed in [3].

### VII. ELICIT REQUIREMENTS

Once the objectives of the system are defined, they need to be operationalised into requirements. Then, an analysis of the contribution that the requirements have to objectives should be performed. This stage is inspired in the *task/function* and *system performance qualities* identification activities of R4IE [5]. Following this, requirements are refined, decomposed into sub-objectives, and related to other model elements. All the decisions taken need to be documented as rationales, in order to facilitate requirements tracing, by modelling the reasons which developers are following to make decisions.

#### A. Refine requirements

Once the engineers agree upon the representation of values, objectives, their decomposition, obstacles and resources; the next step is to discover alternatives which can satisfy the objectives, finding their possible operationalisations, in the form of requirements. In the previous sub-activity, higher-order requirements are identified, as well as their contribution to the objectives. In this sub-activity, those requirements are refined into more precise requirements, and are related to other elements of the system. In addition to those relationships (RefineObj and Contribute) introduced by the Interdependency Graph in REUBI [3], the Requirements Diagram inherits five different types of relationships from SysML (Derive, Refine, Satisfy, Verify and Copy) [7]. This sub-activity mainly consists of applying these seven relationships between requirements, objectives and other elements of the system. More information on the application of these relations can be found in [3], [7].

#### B. Argument decisions

Once the requirements are operationalised and refined, the aim is to model the decisions taken during the previous activities. The Requirements Diagram enables this through the use of the Argumentation stereotype, which is related to other elements in the Requirements Diagram. SysML already provides a means to argument relationships via the Rationale stereotype. Nevertheless, the Argumentation stereotype provided in the Interdependency Diagram of REUBI [3], facilitates the specialisation of the rationale into support and rejection arguments.

#### C. Requirements diagram

For the purpose explained in this section, the Requirements Diagram is introduced, which inherits the stereotypes of the OMG SysML Requirements Diagram [7], and the object and justification meta-models from the REUBI Interdependency Graph [3]. The Operationalisation stereotype from REUBI,

has been substituted by the SysML Requirement, which gives several advantages. The requirements traceability relationships help keep track of what happens to a requirement during system modelling and specification by identifying sources, destinations and links between requirements and models. Additionally, the SysML requirements enable a mapping to evaluation constructs such as the test case, providing a way of documenting how the requirements will be tested, which can be used along with other UML-based standards such as the UML 2.0 Testing Profile [8], to facilitate the design and automation of test runs [21]. SysML also offers two requirements visualisation mechanisms to identify, prioritise and improve requirements traceability through requirements tables and requirements traceability matrixes. Although the exclusive use of Use-case diagrams might be limited for the requirements engineering process, the use of SysML requirements to complement them represents an advantage and improves standardisation [22]. SysML requirements can also be related to use-cases with the *refines* relationship. Finally, the approach also inherits some advantages from the use of REUBI objectives, as these are used to facilitate the discovery of requirements and non-functional requirements and act as a bridge between the stakeholder analysis and the requirements. Finally, it is also important to mention that the operationalisation of objectives into requirements can be evaluated using the evaluation procedures from REUBI [3].

### VIII. EVALUATE

The last activity of the framework consists of an evaluation of the objectives and requirements, which is guided by a set of heuristics, that have been adapted to their application to the framework presented in this paper from [19] [3]. Then, a plan for evaluating the requirements is created, setting the criteria for how each requirement will be evaluated once the system is implemented. For this, the objectives need to be prioritised, in order to enable developers to focus on the development efforts on the most important objectives first. Then, an evaluation helps engineers to determine if the current modelled operationalisation of objectives into Requirements satisfies the objectives. For this the evaluation procedure for the NFR framework is adopted, as presented in [19] and [3].

### IX. CASE STUDY

The case study introduced in this section is based on the insights gained during the development of the EU funded POSEIDON<sup>4</sup> project [23]. The project name stands for Personalised Smart Environments to increase Inclusion of people with DOWn's syndrome, and is particularly focused on using smart assistive technologies in order to foster the independence of people with this condition. The example presented in this work is constrained to an outdoors navigation application, which is bespoke to this particular disability [25]. More specifically, the case study focuses on a mobile application that uses a real-world representation of maps along with location

<sup>4</sup><http://www.poseidon-project.org/>

services to support outdoor journeys that might be walking or by bus. Due to space restrictions, this example is further limited to bus displacements happening in London, United Kingdom. The application uses routes with tailored directions, notifications, reminders, and other services which will be triggered depending on the context. The navigation system described in this case study can be found in [26], and it has been developed using the open-source framework RCASE [10] developed as part of the contribution presented in this paper. All the figures appearing in the remaining of the paper are screenshots of the RCASE tool.

### A. Stakeholder analysis

The stakeholder identification activity presents a set of techniques that build on the previous activity. The following list of stakeholders are identified: 1) Primary Users (PU), people with Down's Syndrome; 2) Secondary Users (SU), parents or carers of people with Down's Syndrome; 3) POSEIDON Managers, the management team of the POSEIDON project; 4) POSEIDON Development Partners, POSEIDON project partners which work in creating code or libraries that are to be reused by this application. 5) Developers, the developers of the navigational system; 6) Bus driver, the person(s) that drive(s) the bus in which the PU will get on; 7) Bus company, the company in charge of the bus line; 8) Calls and Internet provider, referring to the company that provides phone calls, SMS and internet to the mobile device; 9) Device Manufacturer, company that manufactures the device; 10) Operating System Developers, group involved in the development of the operating system of the device; 11) Maps Library Developers, group involved in the development of the maps libraries. The list is further refined into the power versus interest grid, which evolves through iterations into the stakeholder influence grid, as it is shown in Figure 3. The stakeholder

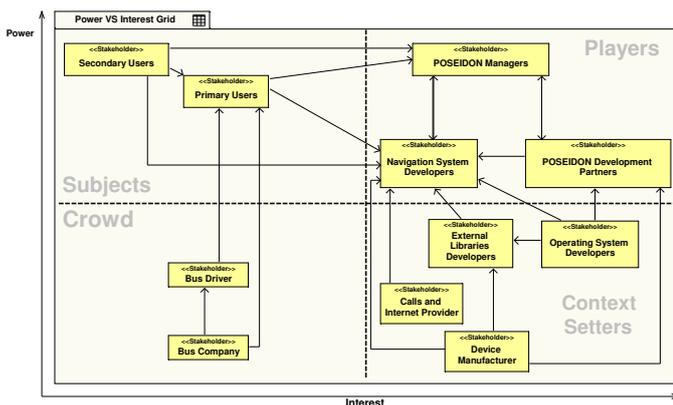


Fig. 3. Power vs Interest Grid representation, created with the Stakeholder Diagram from the RC-ASE Tool.

diagrams introduced give better insights about who are the stakeholders of the system and their relevant aspects to the project. The analysis on the stakeholders and their profiles can provide relevant information of the stakeholders which can be later reused for identifying their needs and preferences

in the context related requirements. The stakeholder profiling activity follows. The POSEIDON project, involved a total of three different countries, namely, United Kingdom, Germany and Norway. These three cultures are similar in the sense of avoiding uncertainty, having similar work patterns, and responding similarly to authority, initiative and responsibility, since they all share the same continent. As expecting users with possibly low-skills with technology [27], the use of visual or symbolic representations of context is preferred, as well as the one task at a time approach. Another relevant aspect to take into account is the language difference between these three countries. Additionally, the United Kingdom has a different currency, representation of metrics, and driving direction than Germany and Norway. This might affect the payments of users for public transport, the location of bus stops, as well as the distance representation in the maps. The discovery of personas revealed that some of the particular users have visual or auditive impairments, and the questionnaires revealed that different skill levels using information technologies [27]. There are five different user profile features for the primary user stakeholder: Culture, visual impairment, skills with technology, independence degree, and auditive impairment. Each of the profile features is divided into its corresponding user profile feature instances. For example, the independence degree can be classified into three profile feature instances: *independent*, *moderately dependent*, and *dependent*. Following, the activity of the end-user stakeholders when displacing is analysed. A user will typically walk to the bus station, wait for a bus, take the bus, press the stop button one destination before the stop, get off the bus and walk again if necessary. This information will be used to identify situational interests in  $F_2$ , as it is shown in Table 1 from [2].

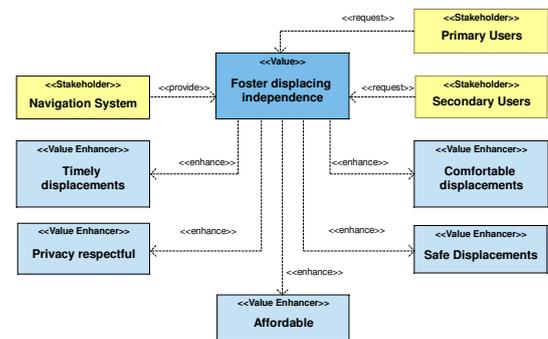


Fig. 4. Value model representation, Stakeholder Diagram, RC-ASE Tool.

The last activity in the stakeholder analysis consists of creating a value model, as shown in Figure 4. The first actor to take place is the navigational system itself, which offers the value of fostering the independence of both primary and secondary users. That value is a service, which is requested by the primary and secondary users. Five main aspects enhance the value provided by the navigation system. These are: that primary and secondary users can afford the system, that the system can preserve the privacy of the users, that the primary users can displace safely when using the system, that the

primary users can reach their destination on time, and that the instructions given by the navigation system are understandable by primary users.

### B. Establish objectives

The main goals and soft-goals of the system are derived from the value model shown in Figure 4. In this way, the goal *Guide displacements*, is related to the *Foster displacing independence* value, as shown in Figure 5. Since this value is still too generic, it needs to be refined. The goal can be decomposed into two sub-goals: *Walking displacement guidance* and *Bus displacement guidance*. Note that for satisfying the high level objective, both lower level objectives must be satisfied. It equally happens with the value enhancers. *Walking displacement guidance* is refined into the objective "Time-based guidance", which proposes that the guidance received by the stakeholders will take into consideration time constraints. This goal is divided into another two lower level goals, which are to *provide guidance about when to start the displacement*, and to *provide guidance according to the waking speed*. The value enhancer *Affordable*, is distilled into the *Low-cost* soft-goal, which at the same time is divided into *Low-cost hardware* and *Low-cost software* soft-goals. The value enhancer *Privacy respectful* is also refined into the soft-goal *User privacy*, that is divided into the two soft-goals *Anonymity/pseudonymity* and *User intimacy*. The value enhancer *Safe displacement* is refined into the soft-goals *Displace through safe environments*, and *Support lost users*. Finally, the value enhancer *Comfortable displacement*, is distilled into the goal *Guide on required objects*, that supports the user with a list of objects that can make more comfortable the displacement or the activity to do where the user is displacing. Also, this value enhancer is refined into the soft-goal *Provide understandable guidance*.

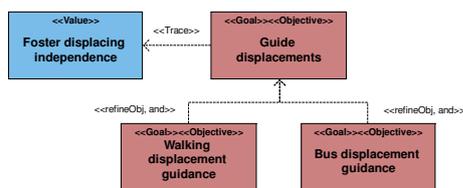


Fig. 5. Goal Decomposition I, Objective Diagram, RC-ASE Tool.

Following, an obstacle analysis over the objectives proceeds. The main obstacle found is due to the interruption of the service, caused by a lack of power. The battery may run off, and the user is left without instructions to follow. In order to mitigate, the soft-goal *Availability* is added. Next, is the Resource analysis. Here, the goal of guiding displacements refines from the *Start instructions* resource, generated from the time-based guidance goal. Additionally, the goal for guiding displacements also requires from the *Personal object list* resource, generated from the *Guidance on object list* goal.

### C. Elicit requirements

At this stage of the method, the different goals of the system are refined into requirements, that represent a condition

or capability that the system needs, and which contribute to the satisfaction of objectives. These design decisions, as well as the positive or negative contributions of the decisions are studied. For this, the lowest-level goals are considered (i.e., those goals which do not have any sub-goal). In the previously introduced goal models, there are 5 low-level goals, which are used to define the functional requirements of the system, and 7 low-level soft-goals, which are used to define the non-functional requirements. For simplicity, the Requirements Diagrams of this example have been divided into three parts: requirements related to navigation, as shown in Figure 6; requirements related to reminders of the system; and non-functional requirements. Navigational requirements are based on a main requirement, *Navigation Map*, that specifies that the user will be able to observe a map that represents the real-world surroundings. Note that this requirement is a positive contribution towards two low-level goals: *Walking displacement guide* and *Bus displacement guide*. However, since just

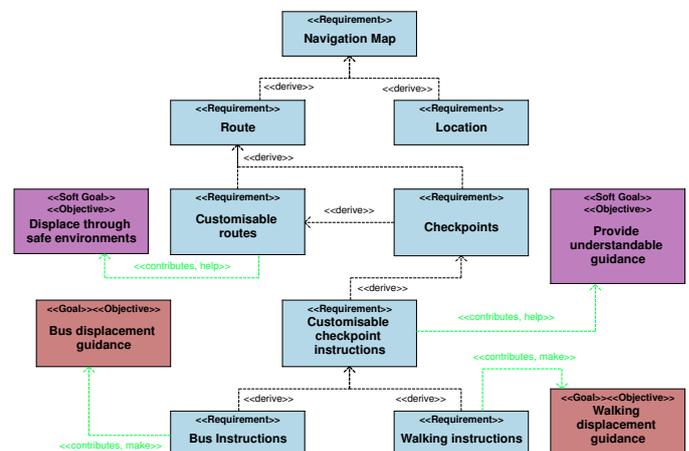


Fig. 6. Requirements model I, Requirements Diagram, RC-ASE Tool.

showing a map can not be considered as providing enough guidance, the contribution relationship can not be considered as a *Make* contribution. To keep the diagram simple, the *help* relationships between requirements and these two goals have been omitted in Figure 6. Since the *Navigation Map* requirement is not enough proof for satisfying the two previously mentioned goals, this main requirement is divided into another two additional requirements which are to show specific instructions on the next movements that users need to do in order to ultimately arrive at their destination. The navigation map will have a *route*, indicating the path that the user has to follow in order to arrive at her/his destination. Additionally, the navigation map will display the *location* of the user in the map in real-time. Although these two new requirements also provide a positive contribution towards the satisfaction of the two main guidance goals of this diagram, they are still not enough proof for providing adequate guidance to the users when walking and displacing by bus. Taking into account the low level objective of *Displacing through safe environments*, the requirement *Customisable routes* is included, where it is

specified that the secondary users will be able to create their own routes for the primary users. The difference between the application under development and other navigation applications, is that this option increases the security of the primary users, as parents are expected to send them through safer and easier routes, instead of the most complicated ones. This requirement also satisfies the needs of users with different skill levels. As it can be observed in Figure 6, this new requirement is considered as a positive contribution towards the soft-goal for safe environments. Routes, will also have *checkpoints* that divide the route into more manageable smaller parts. Although this requirement by itself does not provide any contribution to the objectives, it is necessary to understand the next requirement that derives from it: *Customisable checkpoint instructions*. The checkpoints of the routes, will not only be located by the secondary users, but they will include a set of personalised instructions about the next movement. For example, it could be “*When you see the blue house with a white door, turn left, using the crosswalk*”. An additional picture of the blue house can be included for making the instruction more clear. This requirement positively contributes to the soft-goal of *Provide understandable guidance*. These customisable checkpoint instructions can map to *walking* or for *bus displacements*. These last two instruction types *make* the main guidance goals. Therefore, the requirements engineers can consider this diagram as finished, and continue with the following diagram. Due to space reasons, reminder-related requirements and non-functional requirements have been omitted from this example. The operationalisation into requirements from objectives will occur similarly to that of navigational requirements.

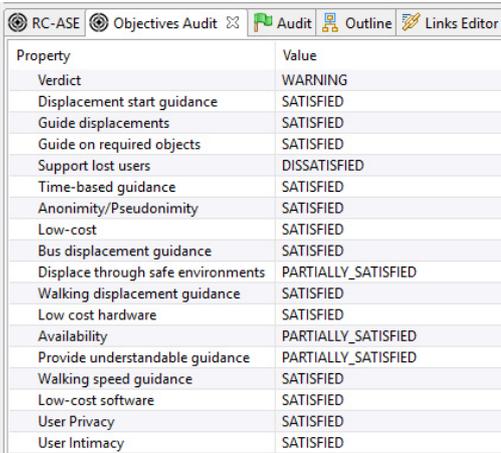
After completing the operationalisation of objectives into requirements, the next step is to personalise or create new requirements according to the different user profiles. As it can be observed in the first figure, the cultural profile affects the existing communication with the users. Therefore, this figure illustrates the different requirements that are created to satisfy the demands of a British profile. The project supports English, German and Norwegian languages, British pounds (GBP) and Euros (EUR), as well as the Imperial and Metric systems. On the other hand, the second figure enables a different communication with the users. For those users with visual impairments, audio based communications will be present, and for those with auditive impairments visual communications will be enabled. The sub-activity for personalisation introduced in Section V-B4 it also includes a specification of the set-up and training. The users of the navigation application, will have to their disposition a training tool for letting them acquire navigation skills in a virtual environment, without exposing themselves to unnecessary risks. For space reasons, the further explanation on the training framework is out of the scope of this example, but the Reader is referred to [28] for more information about how users can train using this system.

Finally, an ethical analysis of the stakeholders is done. For example, an ethical analysis of the primary user stakeholder can be conducted against the *Login*, *Mobile platform*,

*Reminders*, *Navigation map*, and *Communication with the users* requirements. For this profile, there is no *dependency* of the stakeholder on the government regarding the mentioned requirements. Also, there is a medium level of *vulnerability* from the users, in case they can get lost by misinterpreting indications. Nevertheless, the *gravity* of this stake is low. There is a high *likelihood* that there will be a remedy for this which will be addressed when creating the context-awareness specialisation of the requirements methodology. There is a medium *risk* to the integrity of the stakeholders, and the *policy impact* is high.

#### D. Evaluate

First of all, if it has not been done already, all the objectives defined in IX-B need to be prioritised. Then, the R-CASE module will automatically give a verdict on the satisfaction of the objectives of the system, according to the algorithm and rules explained in Section VIII. For this example, the result of this evaluation can be observed in Figure 7. The current verdict of the example is *WARNING*, as the *Support lost users* objective is *DISSATISFIED*, and the *Availability* objective is *PARTIALLY\_SATISFIED*. This means that in order to improve the verdict to *PASS*, special attention should be payed to the completion of these objectives when eliciting requirements related to the context-awareness of the system.



Property	Value
Verdict	WARNING
Displacement start guidance	SATISFIED
Guide displacements	SATISFIED
Guide on required objects	SATISFIED
Support lost users	DISSATISFIED
Time-based guidance	SATISFIED
Anonymity/Pseudonymity	SATISFIED
Low-cost	SATISFIED
Bus displacement guidance	SATISFIED
Displace through safe environments	PARTIALLY_SATISFIED
Walking displacement guidance	SATISFIED
Low cost hardware	SATISFIED
Availability	PARTIALLY_SATISFIED
Provide understandable guidance	PARTIALLY_SATISFIED
Walking speed guidance	SATISFIED
Low-cost software	SATISFIED
User Privacy	SATISFIED
User Intimacy	SATISFIED

Fig. 7. Screenshot of the evaluation of objectives using the RC-ASE module in Modelio.

## X. CONCLUSIONS AND FUTURE WORK

This paper proposes a framework for facilitating the systematic treatment of requirements, and which is specialised for the non-contextual aspects of C-AS. The framework proposes a guide for developers that spans from the identification of stakeholders, to the identification of objectives and its operationalisation of goals, and introducing a UML/SysML profile for supporting the documentation and modelling of the process. The process is based on the strong points of different methodologies which are gathered as a coherent framework, helping to cover the gaps in the development of C-AS that current requirements elicitation methodologies have (Table I).

Additionally, the framework has been implemented as part of an open-source tool which supports the Diagrams introduced in this paper, as well as other SysML features to increase the traceability of elements throughout the models. A novel module for Modelio has been developed, namely Requirements for Context-Aware Systems Engineering (RC-ASE) [10], which implements not only the Diagrams introduced during this section, but also the missing SysML features that the free version has, including *traceability matrixes* and *requirements tables*, as well as other relevant functionality such as partial documentation generation. The approach has been applied to a navigation system of the POSEIDON project. Currently, there is undergoing work to create a more specialised UML/SysML profile that is more focused on the contextual aspects, related to  $F_2$ , as introduced in [2]. More work is being dedicated to the creation of another framework that facilitates the design and automatic code generation, aimed for the management of context information for context-aware rule-based reasoning support in both mobile [29] and stationary [30] platforms. The aim is not only to create services that can create C-AS that are more related to the preferences and needs of the users, but to create more reliable services by automating the verification of reasoning rules.

#### ACKNOWLEDGMENT

The research leading to these results has been partly supported by the POSEIDON project funded by the European Union (FP7/2007-2013) under grant agreement no. 610840.

#### REFERENCES

- [1] U. Alegre-Ibarra, J. C. Augusto, and T. Clark, "Engineering context-aware systems and applications: A survey," *Journal of Systems and Software*, vol. 117, pp. 55–83, 2016. doi: 10.1016/j.jss.2016.02.010
- [2] U. Alegre-Ibarra, J. C. Augusto, and C. Evans, "Perspectives on engineering more usable context-aware systems," *Journal of Ambient Intelligence and Humanized Computing*, 2018. doi: 10.1007/s12652-018-0863-7
- [3] T. Ruiz-López, M. Noguera, M. J. Rodríguez, J. L. Garrido, and L. Chung, "Reubi: A requirements engineering method for ubiquitous systems," *Science of Computer Programming*, vol. 78, no. 10, pp. 1895–1911, 2013. doi: 10.1016/j.scico.2012.07.021
- [4] A. Sutcliffe, S. Fickas, and M. M. Sohlberg, "Pc-re: a method for personal and contextual requirements engineering with some experience," *Requirements Engineering*, vol. 11, no. 3, pp. 157–173, 2006. doi: 10.1007/s00766-006-0030-0
- [5] C. Evans, L. Brodie, and J. C. Augusto, "Requirements engineering for intelligent environments," in *Intelligent Environments (IE), 2014 International Conference on*. IEEE, 2014. doi: 10.1109/IE.2014.30 pp. 154–161.
- [6] OMG, "OMG Universal Modeling Language (UML), Version 2.5," Object Management Group, Tech. Rep., 2015. [Online]. Available: <http://www.omg.org/spec/UML/About-UML/>
- [7] —, "OMG Systems Modeling Language (OMG SysML), Version 1.3," Object Management Group, Tech. Rep., 2012. [Online]. Available: <http://www.omg.org/spec/SysML/1.3/>
- [8] —, "UML 2.0 Testing Profile, Version 2.0," Object Management Group, Tech. Rep., 2017. [Online]. Available: <http://www.omg.org/spec/UTP/>
- [9] R. Fuentes-Fernández, J. J. Gómez-Sanz, and J. Pavón, "Understanding the human context in requirements elicitation," *Requirements engineering*, vol. 15, no. 3, pp. 267–283, 2010. doi: 10.1007/s00766-009-0087-7
- [10] U. Alegre-Ibarra, "Requirements for context-aware systems engineering (rcase) tool," <https://github.com/ualegre/rcase>, [Online; Last accessed 19-February-2018].
- [11] K. Oyama, H. Jaygarl, J. Xia, C. K. Chang, A. Takeuchi, and H. Fujimoto, "Requirements analysis using feedback from context awareness systems," in *Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International*. IEEE, 2008. doi: 10.1109/COMP-SAC.2008.239 pp. 625–630.
- [12] S. Greenberg, "Context as a dynamic construct," *Human-Computer Interaction*, vol. 16, no. 2, pp. 257–268, 2001.
- [13] A. Finkelstein and A. Savigni, "A framework for requirements engineering for context-aware services," in *In Proc. of 1st International Workshop From Software Requirements to Architectures (STRAW)*, 2001. doi: 10.11648/j.ajsea.20150406.11 pp. 200–1.
- [14] W. Sitou and B. Spanfelner, "Towards requirements engineering for context adaptive systems," in *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International*, vol. 2. IEEE, 2007. doi: 10.1109/COMP-SAC.2007.223 pp. 593–600.
- [15] N. Seyff, F. Graf, P. Grünbacher, and N. Maiden, "Mobile discovery of requirements for context-aware systems," in *Requirements Engineering: Foundation for Software Quality*. Springer, 2008. doi: 10.1007/978-3-540-69062-7\_18 pp. 183–197.
- [16] L. Baresi, L. Pasquale, and P. Spoletini, "Fuzzy goals for requirements-driven adaptation," in *International Requirements Engineering Conference (RE), 2010*. IEEE, 2010. doi: 10.1109/RE.2010.25 pp. 125–134.
- [17] J. M. Bryson, "What to do when stakeholders matter: stakeholder identification and analysis techniques," *Public management review*, vol. 6, no. 1, pp. 21–53, 2004. doi: 10.1080/14719030410001675722
- [18] S. Jones, S. Hara, and J. Augusto, "e-friend: an ethical framework for intelligent environment development," in *Ethics and Information Technology*, vol. 17. Springer, 2015. doi: 10.1186/2192-1962-3-12 pp. 11–25.
- [19] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-functional requirements in software engineering*. Springer Science & Business Media, 2012, vol. 5.
- [20] D. Maulsby, S. Greenberg, and R. Mander, "Prototyping an intelligent agent through wizard of oz," in *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. ACM, 1993. doi: 10.1145/169059.169215 pp. 277–284.
- [21] M. Hause, A. Stuart, D. Richards, and J. Holt, "Testing safety critical systems with sysml/uml," in *Engineering of Complex Computer Systems (ICECCS), 2010 15th IEEE International Conference on*. IEEE, 2010. doi: 10.1109/ICECCS.2010.59 pp. 325–330.
- [22] M. dos Santos Soares and J. Vrancken, "Requirements specification and modeling through sysml," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*. IEEE, 2007. doi: 10.1109/IC-SMC.2007.4413936 pp. 1735–1740.
- [23] J. C. Augusto, T. Grimstad, R. Wichert, E. Schulze, A. Braun, G. M. Rødevand, and V. Ridley, "Personalized smart environments to increase inclusion of people with down's syndrome," in *International Joint Conference on Ambient Intelligence*. Springer, 2013. doi: 10.1007/978-3-319-03647-2\_16 pp. 223–228.
- [24] "Official website of the poseidon project," <http://www.poseidon-project.org/>, [Online; Last accessed 19-February-2018].
- [25] D. Kramer, A. Covaci, and J. C. Augusto, "Developing navigational services for people with down's syndrome," in *Intelligent Environments (IE), 2015 International Conference on*. IEEE, 2015. doi: 10.1109/IE.2015.26 pp. 128–131.
- [26] D. Kramer and Tellu, "Poseidon application," <https://play.google.com/store/apps/details?id=no.tellu.poseidon>, [Online; Last accessed 19-February-2018].
- [27] POSEIDON Project, "Poseidon deliverable 2.1 - report on requirements," *PersOnalized Smart Environments to increase Inclusion of people with DOWns syNdrome*, Tech. Rep., 2015. [Online]. Available: <http://www.poseidon-project.org/wp-content/uploads/2015/12/D2.1-Report-on-requirements-revised-after-pilot-without-interviews.pdf>
- [28] A. Covaci, D. Kramer, J. C. Augusto, S. Rus, and A. Braun, "Assessing real world imagery in virtual environments for people with cognitive disabilities," in *Intelligent Environments (IE), 2015 International Conference on*. IEEE, 2015. doi: 10.1109/IE.2015.14 pp. 41–48.
- [29] D. Kramer and J. C. Augusto, "Supporting context-aware engineering based on stream reasoning," in *International and Interdisciplinary Conference on Modeling and Using Context*. Springer, 2017. doi: 10.1007/978-3-540-88479-8\_37 pp. 440–453.
- [30] U. Alegre-Ibarra, J. C. Augusto, and A. Aztiria, "Temporal reasoning for intuitive specification of context-awareness," in *Intelligent Environments (IE), 2014 International Conference on*. IEEE, 2014. doi: 10.1109/IE.2014.44 pp. 234–241.