DOI: 10.15439/2017F272
ISSN 2300-5963 ACSIS, Vol. 11

Development of Simulations for Ambient Assisted Living through Pattern Repositories

Rubén Fuentes-Fernández
GRASIA (Research Group on
Agent-based, Social & Interdisciplinary Applications),
Universidad Complutense de Madrid
in Madrid, Spain
Email: ruben@fdi.ucm.es

Jorge J. Gómez-Sanz
GRASIA (Research Group on
Agent-based, Social & Interdisciplinary Applications),
Universidad Complutense de Madrid
in Madrid, Spain
Email: jjgomez@fdi.ucm.es

***Abstract*—Ambient Assisted Living (AAL) pursues providing an autonomous and satisfactory life to people through technology, independently of their actual conditions. Its developments usually require testing protoypes with real users in Living Labs (LL). This makes projects expensive. Virtual LLs (VLLs) try to address theses issues by using simulations for requirements elicitation and the initial testing of solutions. These simulations frequently require considering social aspects, e.g. relationships, culture, or decision making. These are recurrent and quite application-independent aspects for AAL. Our work proposes *social properties* as patterns that represent these aspects and that can be plugged-in in simulations. The knowledge for these properties is extracted following the Activity Theory (AT) paradigm from Social Sciences. Their specification uses models and transformations (e.g. to generate other models or code) following Model-Driven Engineering (MDE) practices. This faciliates their understading and use in simulation development. A case study on AAL for ageing illustrates the approach.**

I. Introduction

AMBIENT Assisted Living (AAL) [1] pursues the development of socio-tecnical solutions that facilitate to people carrying out an independent and satisfactory life, regardless their particular conditions, both physical and mental. Its solutions are intended to support a variety of needs, like those of elderly people, support for temporal or permanent impairment, or home safety and security.

AAL solutions are socio-technical systems [2], which depend on human, social, and organisational factors, technical features, and their interplay. The design of such systems fall beyond traditional practices of system development, as it needs to take a closer, holistic, and pluri-disciplinary look to all these elements [3]. In the case of AAL, design needs to consider human aspects such as what assistive technologies people accept, the takecarers' involvement, or the users' self-image.

The discovering and testing of these requirements frequently demands the use of Living Labs (LLs). A LL [4] is an open research and innovation ecosystem. It gathers the different stakeholders (e.g. user communities, developers, researchers, policy makers, and investors), and the resources needed for their interaction. In particular, it usually includes the settings and devices to analyze needs and technologies, and to evaluate and test hypotheses, technologies, and solutions. The main issues with these facilities are that they are expensive (to set up and keep updated, and therefore to use), and that they strongly constrain tests (e.g. only with available devices, in real time, and controlling potential damages). In turn, this highly increases the cost of AAL solutions.

The research on Virtual LLs (VLLs) [5] addresses these issues through the development of very accurate software simulations of LL. The AAL system is deployed in the VLL as it would be in a LL. The VLL receives its outputs and actions and provides the relevant inputs. From the perpective of the AAL system, there is no difference between deployment in a LL and a VLL. The VLLs can replace physical LLs in the early stages of the development of solutions, mainly regarding requirements elicitation and early prototyping. Of course, VLLs have limitations. It is difficult to consider any potential event and interaction that can appear in the real world, but a VLL can still consider a wide variety of them.

Social aspects have a pervasive impact on AAL system, that frequently affects them in quite abstract aspects, like acceptance, interaction with devices and other users, or concerns on privacy. This high level of abstraction facilitates its reuse in different contexts. At the same time, it frequently puts them far beyond the usual background of researchers and developers from technical-oriented fields.

The previous considerations took our research to consider the need for VLL, and for extension in AAL, of ready-to-use knowledge on social aspects for the design of these systems. This knowledge should alleviate the workload on eliciting and applying these aspects in the development of AAL solutions.

Our work proposes the use of *social properties* [6] and Model-Driven Engineering (MDE) [7] to address these issues. Social properties crystallize knowledge from Social Sciences in forms useful for development, that MDE techniques can quickly incorporate to simulations models and code.

Social properties [6] are similar to design patterns [8] in Software Engineering, i.e. they are templates of general solutions to problems that repeteadly appear in the development of systems, and are described at different abstraction levels (from graphical specifications to code). When available, engineers use and combine them to address well-identified issues of their systems. The aplication of patterns is usually a manual task, i.e. engineers design / write how the pattern is applied.

IEEE Catalog Number: CFP1785N-ART ©2017, PTI

However, in the context of MDE [7] this application can be semi-automated.

MDE [7] is an approach to system development organized around *models* and *transformations*. It formally defines Modelling Languages (MLs) that are domain-specific. Engineers specify their systems using *models* that conform to these MLs. *Transformations* [9] allow generating models from models (M2M) (e.g. addition of platform-oriented information and refactoring.), text from models (M2T) (e.g. documentation and code generation), or models from text (T2M) (e.g. reverse engineering). The development process is then conceived as an iterative refinement of model specifications through the manual addition of infomation and the semi-automated transformation of models.

In a MDE context, social properties are defined using models and transformations. Having specifications at different level of abstraction allows addressing the need of high-level design with domain expert, but also of developers in late design. Moreover, making explicit all this information facilitates the traceability of all the artefact in developement. Therefore, it helps to guarantee that the simulation corresponds to the initial requirements.

The last element of the approach is related to how to obtain useful knowledge from Social Sciences to design AAL simulations in VLLs. Our work resorts to Activity Theory. The Activity Theory (AT) [10] is a paradigm for interdisciplinary research based on a socio-cultural approach. Previous works [11], [12] have proven the advantages of AT in the development of Multi-Agent Systems (MAS) as a source of expertise about intentional and social aspects. In particular, this knowledge has been used in requirements elicitation to study interactions among systems and their users in an integrated way [12].

The paper illustrates the application of the approach with a case study about the development of an AAL solution with different types of users. These users experience some problems when moving (e.g. falls, blockades, or inability to grasp objects). However, their features regardingg age, gender, cultural background, or familiar relationships are different. The simulation needs to consider these variations to study the feasibility of a surveillance and help request system for these people at their homes. Thes case study shows how social properties consider these features and introduce them in the simulation running in a VLL.

The remainder of the paper is organized as follows. Section II presents AT. Section III describes *social properties* and section IV the process to use them in simulations, which section V applies in the case study. Section VI compares the presented approach with existing AAL works. Finally, section VII discusses some conclusions and future work.

II. Activity Theory

The Activity Theory (AT) [10] is a socio-psychological framework for the study of human behaviour. It focuses on the dialectics between people and their physical and social environment. The enviroment shapes human actions and their execution, but at the same time, these actions also partially define and change the enviroment. This interaction occurs over time. Since these perspectives are inherently interleaved, AT advocates for their holistic analysis. These acts in context constitute the minimal meaningful unit of analysis and are called *activities*. AT makes no distinction between physical and mental activities.

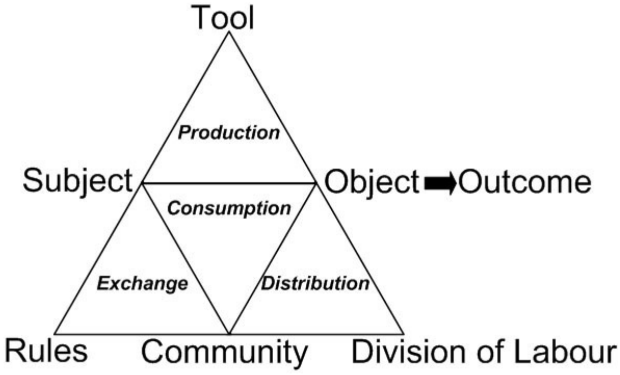

Fig. 1. AT depiction of an activity system [13].

An *activity* [13] is a transformation process driven by people' needs. The *outcome* of the activity is a product able to satisfy those needs. The activity produces the outcome through the transformation of the initial *objects*. The *subject* is the active element that carries out the activity. Any resource the subject uses is a *tool*.

Subjects with a set of common social artefacts constitute a *community*. It characterizes the socio-historical context. The relationships in the community and of this with other elements are mediated by two types artefacts: *rules* and *division of labour*. They include elements such as norms, tacit knowledge, or learned behaviours. The key difference is the focus. The division of labour is related to the organization of the community in the activity. It includes aspects such as goals, hierarchies, collaborations, or responsibilities. Rules represent the context that comes from outsdide the scope of the activity but affects it. They include, for instance, religious beliefs, state laws, or socially acceptable behaviours.

All these elements make up the *activity system*, i.e. the context of an activity. Fig. 1 shows its traditional depiction as introduced in [13].

Activity systems do not appear isolated. They are always interconnected with neighbour activity systems through shared artefacts. For instance, the execution of an activity produces outcomes that become the artefacts (e.g. subject, object, or division of labour) of other activities.

AT also considers the hierarchy of subjects' motives, based on their relevance and how conscious subjects are about them. *Activities* are linked to the high-level *objectives* that they are potentially able to satisfy. *Objectives* meet people needs. *Activities* are executed through networks of *actions*, i.e. sequences of *actions* with alternatives. *Actions* pursue low-level *goals*, which decompose the *objectives*. Subjects are also conscious about goals. At the lowest level, *actions* are implemented through *operations*. These depend on the specific state of the *environment*. Operations are frequently internalized by subjects, so they are unconscious about the actual steps to execute them.

The evolution of activity systems over time depends on their inner *contradictions*. These contradictions are conflicts between the elements in the networks of activity systems. There are four levels of contradictions according to where they appear. Primary and secondary contradictions appear inside an activity system. Primary ones happen in an artefact, or between artefacts of the same type. For instance, because some tools are not designed to work together. Secondary contradictions appear between artefacts of different types. For instance, because tools are not suitable to transform the object. Tertiary contradictions appear between different states of the historical evolution of an activity system. For instance, the care systems for elderly people based on the family and the more recent ones based on technology and hired assistance. Finally quaternary contradictions appear among different activity systems. For instance, a system produces an outcome that is not a suitable tool for another system.

Subjects try to remove contradictions through the evolution of the involved activity systems. This usually generates other contradictions, which produce further evolution of systems.

III. Social Properties

The specification of AAL solutions to analyse their human and social aspects is a demanding task. In order to reduce this effort, our work tries to take advantage on the similarities between different scenarios. For instance, the attitudes towards privacy, surveillance, and technological skills can be common to multiple scenarios. The concept of *social property* aims at describing these reusable partial specifications.

A *social property* specifies a human aspect recurrent in different scenarios of socio-technical systems. The use of these properties in AAL pursues multiple goals: to document a social aspect; to facilitate communication among stakeholders; and to support the development of solutions. To meet all these objectives, social properties are described with the structure represented in Fig. 2.

A *social property* has a unique identifier and a description. The description is a general explanation of its meaning, the context where it can be appplied, and its effects.

The *settings* provide the detailed specification of the property. A *setting* is a specific application of a property in a type of context, i.e. a kind of social group and environment. For instance, a social property describes a structure that relates caretakers and caregivers when the later are hired staff, and its settings account for the differences between organizations with these people in the same premises or caregivers using systems for remote monitoring.

The description of a *setting* includes text and models. UML-AT [11] is the ML used to describe these models. It is a Unified Modelling Language (UML) [14] profile that represents the main concepts of AT (see Section II) with extensions like inheritance and decomposition relationships, constraint expressions, and the concept of *artefact*. The properties in models can be constant or variables to ground when the setting is applied in a simulation. Settings may have *examples* that hep to understand their application.

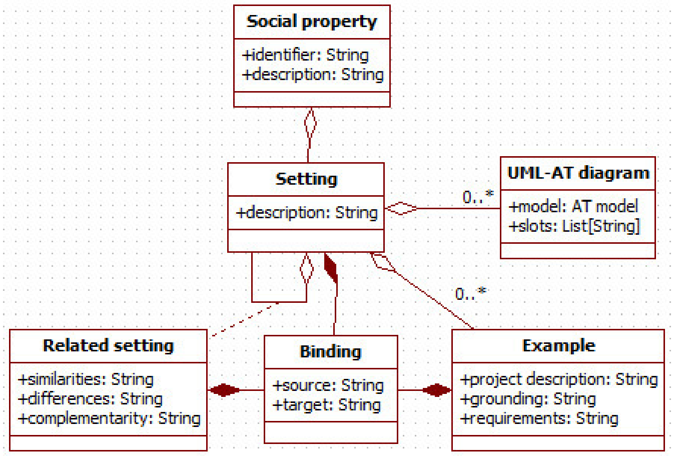

Fig. 2. Structure to describe social properties.

The *related settings* link settings from different *social properties* whose contexts or artefacts are related. For instance, settings about familiy, care organization, and normative frameworks can be related. The *related settings* also discuss similarities and differences among linked settings.

Finally, the *bindings* are pairs of names that associate *settings* or *settings* with *examples*. They indicate which pairs of variables or variable-value must be mapped between the settings or with the example. For instance, they indicate that the outcome of an activity is the tool in a related setting.

As the introduction discussed (see Section I, the settings of a social property can correspond to different levels of detail (e.g. abstract requirements or target simulation platform), contexts (e.g. cultures or countries), and targeted to different transformations (e.g. simulation platforms or documentation). These properties assist in the specification of AAL simulations with predefined parts of models and suggest potential alternatives for model refinement.

IV. Using Social Properties

The *social properties* of the previous section provide the reusable social knowledge to add new information to AAL specifications in a semi-automated way. The activity diagram in Fig. 5 depict this process.

Task 1 *Check applicability* considers whether a social property can be applied in a given context through its *settings*. The specification of a setting includes a description, models, and transformations that can be checked against the current specifications. Usually, a setting can be applied to some specifications to introduce a complete set of new entities. If the setting is going to be connected to some elements already existing in the specifications, some additional checkings are needed, considering that variables can match any value.

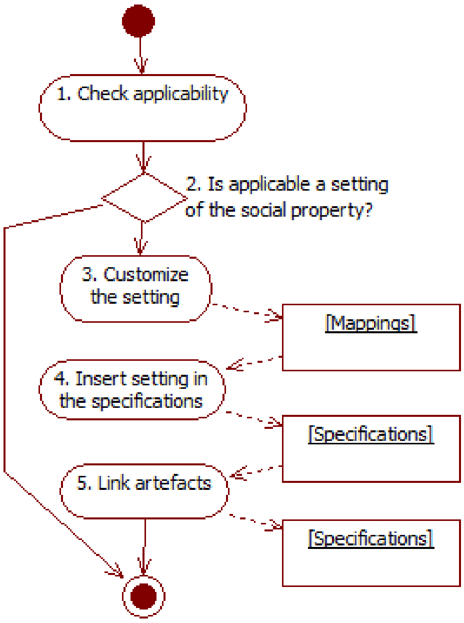

Fig. 3. Process to apply the *social propeties*.

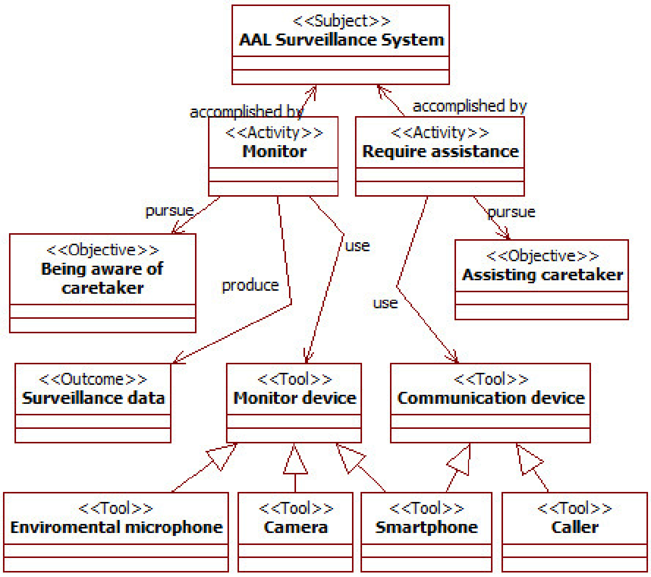
<<Subject>> AAL Surveillance System
accomplished by
accomplished by
<<Activity>> Monitor
<<Activity>> Require assistance
pursue
pursue
<<Objective>> Being aware of caretaker
<<Objective>> Assisting caretaker
produce
use
use
<<Outcome>> Surveillance data
<<Tool>> Monitor device
<<Tool>> Communication device
<<Tool>> Enviromental microphone
<<Tool>> Camera
<<Tool>> Smartphone
<<Tool>> Caller

Fig. 4. Enviroment of the AAL system.

In task 3 *Customize the setting*, the engineer chooses the values to instantiate the non-grounded variables. These values can already appear in the specifications or the user can directly provide them.

Task 4 *Insert setting in the specifications* adds the customized setting to the AAL specification. This is usually done through the setting transformations, modified for the specific context.

In task 5 *Link artefacts*, engineers can introduce additional relationships to connect artefacts. For instance, they can specify that an added artefact is a subtype of an existing one.

Transformations are not only used to add information to specifications. Some social properties represent features to keep or avoid in specifications. For instance, properties can represent AT contradictions (see section II). In that case, running setting transformations over specifications helps to check them.

V. Case study: AAL for People with Motor Problems

The attitude towards technology depends on many factors. The level of adoption of mobiles technologies regarding age is a classical one [15], [16]. The design of AAL systems must consider these variations in order to provide effective solutions. This case study considers a monitoring system and how it can be adapted to the age of its final users using social properties.

The proposed monitoring system has two main functionalities: monitor and ask for assistance. The first one watches the caretaker user and tries to determine when the caretaker may need assistance. In this last case, it uses some communication device to call the caregiver.

In order to support the previous functionality, the system has several devices. It can run a specific app in the caretaker's smartphone. The sensors in the smartphone can provide information on the state of its user. Alternatively, it can use some environmental microphones and cameras to gather information. Microphones are sometimes less reliable, particularly in noisy situations. Cameras require a higher computational capacity than the other devices to process their video.

Regarding users, elderly people are less prone to use smartphones, while youngsters always keep them close and use them continuously. Moreover, youngsters are usually more comfortable with being under video surveillance than elders, that perceive it as highly intrusive. The attitude towards microphones is worse in youngsters than in elder, as the first group is more conscious about the capabilities of modern devices.

This case study is going to use the UML-AT language (see Section II) to model all these elements. The interested reader can find a more complete description of the language in [6], [11].

Let start modelling the AAL solution. From the point of view of AT, the *AAL Surveillace System* is a *subject* able to carry out two *activities*: *monitor* and *require assistance*. The purpose of executing these *activities* is represented with the *objectives being aware of caretaker* and *assisting caretaker*. In these *activities*, it uses two types of device that are *tools*: some *monitor devices* to watch the caretaker, and *communication devices* when it needs to call the caregiver. Given its capabilities, the smartphone is the only device that belongs to both categories. Fig. 4 shows this information.

In this case, the *social properties* report the information on the usual preferences of users. For the sake of simplicity,

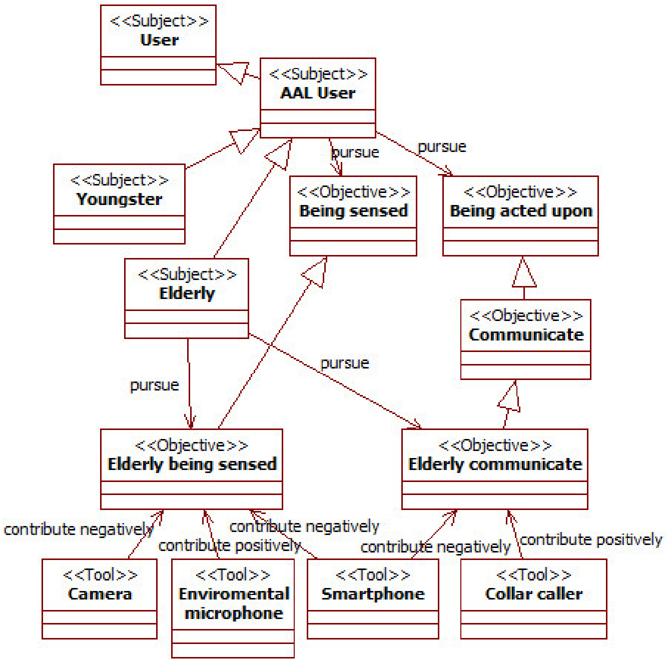

Fig. 5. *Social property* for AAL users: models of *settings*.

the case study considers one *social property* with two *settings* that summarize the usual preferences of elderly and youngster users. In UML-AT, both *users*, *elderlies*, and *youngsters* are *subjects*. In relation to AAL solutions, these users have two main objectives: *being sensed* and *being acted upon*. *Communicate* is modelled as an *objective* sub-type of *being acted upon*. Fig. 5 shows this information.

The preferences of users on devices are represented using contribution relationships on *objectives* (see Section II). These relationships allow specifying when an artefact (i.e. any AT element) helps or damages the possibilities of achieving the purpose of other artefacts. Fig. 5 describes how users prefers interacting with different devices according to their age. The models of both settings for elderly and youngster are combined to simplify their discussion.

The objectives *being sensed* and *being acted upon* are subtyped in order to acknowledge how different *subjects* have different preferecnes regarding their satisfaction. In the case of *elderly subjects*, they prefer trying to achieve the *objective elderly being sensed* using the *tool enviromental microphone* (relationship *contribute positively*) instead of the *tools camera* or *smartphone* (relationship *contribute negatively*). In the same way, for the *objetive elderly communicate*, it is preferred to use a *collar caller* that they usually carry with them, instead of the *smartphone*, that they take with them less and find more difficult to use.

The diagram does not include the preferences of the *youngster subject*. That part of the model is similar to the one discussed for the *elderly subject*.

The application of the *social property* in Fig. 5 to the AAL solution in Fig. 4 follows the process described in section IV. The three steps are as follows for the elderly setting.

Task 1 *Check applicability* consider whether the social property can be used in the context of the current specifications. Here it is used to add information on the users' preferences as new model elements. There is no constraint that precludes is appication.

In task 2 *Customize the setting*, engineers decide how to map those elements that are common to the specifications and the property. The mapping here just indicates that the *tools* that appear with the same name in the specifications and the property are the same artefacts, e.g. *camera* and *smartphone*.

Task 4 just merges both models using transformations. This adds the entities and relationships from the setting to the specifcation of the AAL system.

Finally, task 5 *Link artefacts*, adds additional relationships among artefacts. In this case, they need to indicate how to meet the users' objectives, so that their experience with the AAL system is positive and they use it. The specifications model this information with new contribution relationships. The user's *objectives elderly being sensed* and *elderly communicate* respectively contributes positively to the AAL system *objectives being aware of caretaker* and *assisting caretaker*.

After introducing this information, engineers can perform an automated analysis of the specifications. Navigating relationships from objectives, they can discover the user preferences. Changing the *setting* to use that for youngster, changes the preferences. This indicates that the AAL system needs to be aware of the age of the user.

VI. Related work

This work is related to several areas of research: studies on people, their features and behaviour linked to socio-technical systems; and the development and simulation of AAL solutions. The first group of works is related to sources of knowledge for *social properties*, and how to model it. The second group is linked to the tradeoffs of using these properties in developemt.

Studies with actual people provide information on the relevant attributes and processes to consider when designing AAL solutions. Some of them (e.g. [17]) are focused on the problems that appear in the daily life of certain population groups and are useful to characterize them. There are also works realted to the attitudes of people towards certain aspects of socio-technical solutions, like design issues [15] or technologies [16].

Those studies are useful sources of information to design AAL solutions. However, engineers must perform an important work to extract that knowledge, adapt it to their context, and transform it to requirements. There are already works intended to provide some design knowledge as reusable patterns, mainly in the wider context of Ambient Intelligence [18], [19]. However, these are more at the level of system design, like traditional design patterns [8].

In this context, social properties are means to reuse social knowledge, and are at a higher level of abstraction than available patterns. Both types of pattern can be used in a complementary way. The use of patterns facilitates reuse and communication, and thus reduces the effort on these development aspects.

Currently, most of AAL developments follow traditional practices (e.g. [1], [20], [21]). Experts and engineers design the solution, then the code and relevant hardware is produced, and finally it is tested. Research [5] has already discussed the problems regarding the high costs and failures of projects following these approaches. The use of VLLs tries to mitigate them. Both with ad without VLLs, some works propose the use of MDE techniques [22]. Working with models facilitates the automated generation of systems using MDE techniques.

Social properties suport all these alternatives ways of development. Settigns can be defined at different levels of abstraction and for different targets. For instance, there can be transformations for several target simulation platforms. Nevertheless, the effort to adapt the property to a specific development can largely vary.

VII. Conclusion

This paper has introduced *social properties* as means to deal with social knowledge in AAL developments, particularly in the context of VLLs. They support works by providing ready-to-use knowledge and ways to integrate it in the technical design.

The knowledge can be extracted from different studies following the paradigm of AT [10]. The paradigm offers the tools to study settings, and previous works help to specify [6] them as properties.

This work also proposes a process to apply the properties. It is based on the use of MDE transformations. These transformations support checking the appearance of properties and adding their information to specifications.

The case study has shown how the simulation of an AAL solution can be tailored to consider different features of its user or their social context. Social properties allowed the quick change of the simulation in order to distinguish elder and young users, and the modifications this brings to the enviroment of the system.

The previous work is still ongoing. The effective application of social propeties requires a relevant number of them to model complex contexts. Moreover, transformations need to be adapted to the MLs of the specifications where they are applied and the target simulation platforms. Work in this line was reported in [11] for UML-AT and MLs for MASs. Second, the definition of complex social structures requires being abble to provide more information on the properties of artefacts. The current language for constraints is based on the Object Constraint Language (OCL) [23], and limited regarding social issues. A tailored domain-specific language is currently being designed. Third, further validation and assesment of the utility of properties in AAL developemt is needed. Experiments until now correspond to research projects in AAL.

Acknowledgment

This work has been done in the context of the projects “Fostering a Transition towards Responsible Research and Innovation Systems (FoTRRIS)” (grant 665906) supported by the European Commission in the Horizon 2020 programme, “Collaborative Ambient Assisted Living Design (ColoSAAL)” (grant TIN2014-57028-R) supported by the Spanish Ministry for Economy and Competitiveness, the research programme MOSI-AGIL-CM (grant S2013/ICE-3019) supported by the Autonomous Region of Madrid and co-funded by EU Structural Funds FSE and FEDER, and the “Programa de Creación y Consolidación de Grupos de Investigación” (UCM-BSCH GR35/10-A).

References

[1] G. van den Broek, F. Cavallo, and C. Wehrmann, *AALIANCE Ambient Aassisted Living roadmap*. IOS press, 2010, vol. 6.

[2] F. W. Geels, *Technological transitions and system innovations: a co-evolutionary and socio-technical analysis*. Edward Elgar Publishing, 2005.

[3] G. Baxter and I. Sommerville, “Socio-technical systems: From design methods to systems engineering,” *Interacting with computers*, vol. 23, no. 1, pp. 4–17, 2011. doi: 10.1016/j.intcom.2010.07.003

[4] M. Pallot, B. Trousse, B. Senach, and D. Scapin, “Living lab research landscape: From user centred design and user experience towards user cocreation,” in *First European Summer School “Living Labs”*. INRIA, 2010, pp. 1–10, inria-00612632.

[5] P. Campillo-Sanchez, J. J. Gómez-Sanz, and J. A. Botía, “Phat: physical human activity tester,” in *International Conference on Hybrid Artificial Intelligence Systems*. Springer, 2013, pp. 41–50.

[6] R. Fuentes-Fernández, J. J. Gómez-Sanz, and J. Pavón, “Modelling culture through social activities,” in *Perspectives on Culture and Agent-based Simulations*. Springer, 2014, pp. 49–68.

[7] R. France and B. Rumpe, “Model-driven development of complex software: a research roadmap,” in *Proceedings of the 2007 Future of Software Engineering Conference (FOSE 2007)*. IEEE Computer Society, 2007. doi: 10.1109/FOSE.2007.14 pp. 37–54.

[8] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design patterns: elements of reusable object-oriented software*. Reading, MA, USA: Addison Wesley Professional Computing Series, 1995.

[9] S. Sendall and W. Kozaczynski, “Model transformation: The heart and soul of model-driven software development,” *IEEE software*, vol. 20, no. 5, pp. 42–45, 2003. doi: 10.1109/MS.2003.1231150

[10] A. N. Leontiev, “Activity, consciousness, and personality,” 1978.

[11] R. Fuentes, J. J. Gómez-Sanz, and J. Pavón, “Integrating agent-oriented methodologies with UML-AT,” in *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*. ACM, 2006. doi: 10.1145/1160633.1160873 pp. 1303–1310.

[12] 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.

[13] Y. Engeström, *Learning by expanding: an activity-theoretical approach to developmental research*. Orienta-Konsultit, 1987.

[14] Object Management Group, “Omg unified modeling language (omg uml) - version 2.5,” Technical Report, 2015.

[15] A. D. Fisk, W. A. Rogers, N. Charness, S. J. Czaja, and J. Sharit, *Designing for older adults: Principles and creative human factors approaches*. CRC Press, 2009.

[16] K. Renaud and J. Van Biljon, “Predicting technology acceptance and adoption by the elderly: a qualitative study,” in *Proceedings of the 2008 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT 2008) on IT Research in Developing Countries: Riding the Wave of Technology*. ACM, 2008. doi: 10.1145/1456659.1456684 pp. 210–219.

[17] European Commission, Directorate-General for Economic and Financial Affairs, “The 2015 ageing report - underlying assumptions and projection methodologies,” Technical Report, 2014.

[18] Z. Pousman and J. Stasko, “A taxonomy of ambient information systems: four patterns of design,” in *Proceedings of the working conference on Advanced visual interfaces*. ACM, 2006. doi: 10.1145/1133265.1133277 pp. 67–74.

[19] H. Nakashima, H. Aghajan, and J. C. Augusto, *Handbook of ambient intelligence and smart environments*. Springer Science & Business Media, 2009.

[20] H. Sun, V. De Florio, N. Gui, and C. Blondia, "Promises and challenges of ambient assisted living systems," in *Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on*. Ieee, 2009, pp. 1201–1207.

[21] M. Vacher, F. Portet, A. Fleury, and N. Noury, "Development of audio sensing technology for ambient assisted living: Applications and challenges," *Digital Advances in Medicine, E-Health, and Communication Technologies*, p. 148, 2013.

[22] M. Memon, S. R. Wagner, C. F. Pedersen, F. H. A. Beevi, and F. O. Hansen, "Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes," *Sensors*, vol. 14, no. 3, pp. 4312–4341, 2014. doi: 10.3390/s140304312

[23] Object Management Group, "Object constraint language - version 2.4," Technical Report, 2014.