Exploring Domain Requirements and Technology Solutions: A Goal Modeling Approach

Davide Calvaresi[1,3], Arnon Sturm[1,2], Eric Yu[1], Aldo Franco Dragoni[3]
University of Toronto[1], Ben-Gurion University of the Negev[2], Università Politecnica delle Marche[3]
imdavide@gmail.com, sturm@bgu.ac.il, eric.yu@utoronto.ca, a.f.dragoni@univpm.it

Abstract. As a requirement engineering technique i* has been used to model requirements for a single system. In this paper, we consider whether i* can be used to explore and map user needs and requirements for an entire application domain rather than for a single system. A domain-wide requirements model can be used to assess the suitability of various technology architectures and solutions for that domain. The domain of Ambient Assisted Living (AAL) is characterized by a large variety of stakeholders with different professional and socio-cultural backgrounds. The domain is highly heterogeneous and thus suitable for our purpose of demonstrating domain exploration. We discuss the challenges in mapping that domain, and our attempts to adapt i* concepts and usage for this purpose.

1. Introduction

i* is used as a requirement modeling technique, usually leading to a specific target system. However, nowadays, systems have become much more integrated attempting to address a wider view of the domains of which the systems are related to and trying to address the various stakeholders' viewpoints. Thus, in this paper, we study a case in which i* is used in a wider form to explore a domain of interest and would provide a way of assessing the strengths and weaknesses of various technology architectures and solutions for that domain. Having an understanding of the needs and requirements of an entire domain would facilitate the choice of architectures and solutions to address specific needs. As an example, we have looked into the evolving and complex application domain of Ambient Assisted Living (AAL) [1]. In this domain, the objective is to provide assistance to patients primarily in their homes. Stakeholders include patients, relatives, social services, health workers, and care agencies. Following the domain objective - to increase quality of life for patients and relatives, it is required to support stakeholder needs such as: the actual treatments, communication, environmental monitoring, health monitoring, first aid provisioning, and privacy management. A great variety of technology solutions have been proposed to support AAL. However, in most proposals, the motivations or justifications for the choices made are unsystematic or sketchy and it is not clear how the proposed solutions actually address stakeholders' needs. Our objective is to provide a method to explore and analyze domain requirements so that one can assess how well stakeholders' needs are met by any proposed architecture or solution without referring to their detailed technical construction. To address this aim, we adopt the Goal-Oriented Requirement Engineering (GORE) approach and in particular i* [2] to map out the requirements for the AAL domain and various proposed solutions. The mapping process results in a graph presenting the needs, the solutions and contribution links showing how each solution contributes to addressing the needs. To further clarify, we propose to **map out** the domains using i* (with some adaptations where necessary), without drilling down into the details of specific solutions. The map is also a structured guide to the source documents that contain the original findings or claims.

Challenges encountered include the determination of the right level of abstraction and the semantics of the contribution links. To address these challenges, we propose some adaptations and refinements on the use of i* for this purpose.

2. Using i* for Mapping a Domain

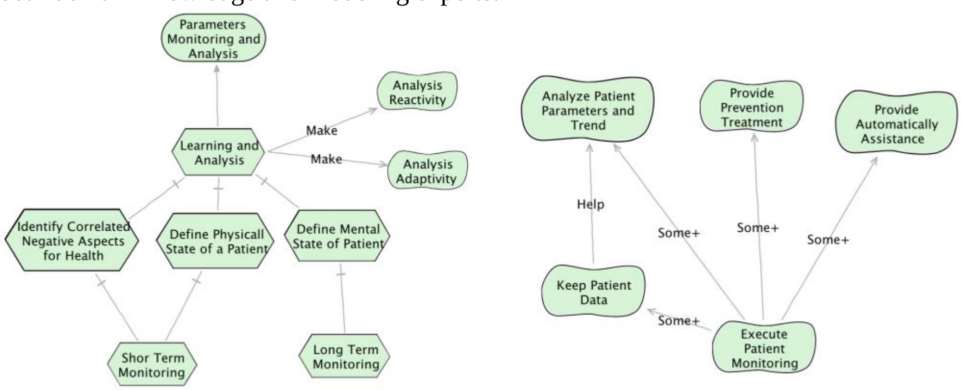
By exploring a domain we aim at performing reflection or reverse engineering from existing evidence (e.g., interviews, existing systems, literature) and thus we call for its mapping. The mapping procedure at this stage is a human-intensive task and none of the activities are yet automated. The mapping of the AAL domain was conducted in five stages.

1. *Gathering requirements*: in this stage, we collected and organized the requirements from scholarly literature about the AAL domain and existing solutions, existing systems (which mainly refer to features the systems provide), and from interviews with representative stakeholders.
2. *Modeling requirement fragments*: in this stage, we modeled requirement fragments in i*. By fragment we mean that we focus on part of the system or needs, rather to capture the whole system/domain in a single model.
3. *Eliciting requirements/needs*: in this stage, we transformed (manually) specific systems i* model fragments into needs model fragments.
4. *Unifying requirement fragments*: in this stage, we unified all requirements and mapped out a first draft of the AAL Domain.
5. *Re-organizing requirement model*: in this stage, we placed each requirement/need/solution to the appropriate level within the graph, where the top levels are need-oriented and lower levels are solution-oriented.

Stage 1 is a preliminary phase of stating the needs, requirements, and solutions.

In stage 2 we modeled the gathered requirements into fragments of i* models. During this process, we clearly observed that the needs/requirements originated from the two types of sources differ in their abstraction level. From the literature on technology solutions, the requirements related to monitoring from the system/implementation point of view are focused on sub-problems and related specific issues such as sensors, scenarios, and task modeling. From the interviews, the requirements were focused on the stakeholders' expected services such as patient's presence at home, patient's movements, and communicating patient's vital parameters.

Thus, in stage 3 in order to further understand the needs from existing systems we have transformed the system models into needs models. In Figure 1 we present an example of such transformation, where in Figure 1(a) we present a fragment of a system model and in Figure 1(b) its transformation into a needs model. The needs (elements at the highest abstraction level) are extracted from the system model and merged with the results of stakeholders interviews. They are modeled as soft goals with their related contribution links. Note that we anticipate the transformation would be done manually and required both domain knowledge and modeling experts.



(a) (b)

Figure 1. Transforming system graph to need graph

To capture the requirements and needs and to understand their relationships, in stage 4 we merged all of the fragments into a unified graph. However, we observed that as the graph is a mixture of various requirements originated from various sources, it is not well organized and it requires further processing. To further organize the graph for the purpose of analyzing the domain, in stage 5 we assigned the various elements into a specific abstraction level by means of clustering, refinement, and manipulation. This assignment process, which is done manually, includes changing or omitting existing nodes (i.e., needs), as well as adding new ones to be better aligned with the new hierarchy. Indeed, these kind of choices are not systematic, and they depend on the domain expert's knowledge regarding the domain and the modeler intuitions of using the modeling language.

We ordered the abstraction levels in a top-down manner, in which the top levels are much related to the needs, and the lower levels are much more related to the actual (technical) solutions. Following our experience, we found out that needs may be captured in many different ways. Sometimes to achieve a specific need from a specific stakeholder view point, it is enough to explicate only the need itself. Alternately, a specific need could require the satisfaction of more than one need or sub-need, to achieve the higher need. For example in Figure 2, the goal "Reduce Treatment Time" can be achieved in different ways: case A represents the patient's view point, case B represents the physician's view point, and case C represents the need "Facilitate Asking help" is achievable in the same way for both patients and physicians. Furthermore, needs at the high abstraction levels are hard to be quantified. For that reason, we chose to specify the higher level needs as i* soft goals even if those are referring to functionality. This also affects the naming of such goals due to the needs' fuzziness in the most upper levels, the semantics ambiguity, and the background that distinguish different domain's points of view. Also, due to different stakeholders' viewpoints, it is difficult to estimate or quantify the effective contribution among needs. In order to address this problem, we used the "some+" contribution link to indicate the existence of such relationships/contributions.

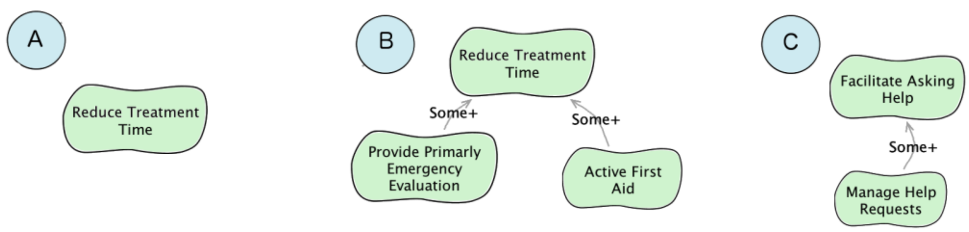


Figure 2. Abstract levels heterogeneity

The assignment to the various abstraction levels introduces two major modeling challenges: grouping the needs and eliminating their redundancies and overlaps. The overlaps are originated by the diversity between several stakeholders' points of view, with which the needs graph was originally built and may result from the assignment process iterations. Nevertheless, when performing these steps, we aimed to keep the original meanings while maintaining the coherency between the different levels. Figure 3 demonstrates assignment of overlapping needs are assigned into a unified model fragment.

The assignment to various abstraction levels introduces another challenge of refining the contribution links. For example, in Figure 4(a) an initial graph is presented. In a later stage, as additional needs are introduced, there is also a need to adjust the contribution links (as can be seen in Figure 4(b)).

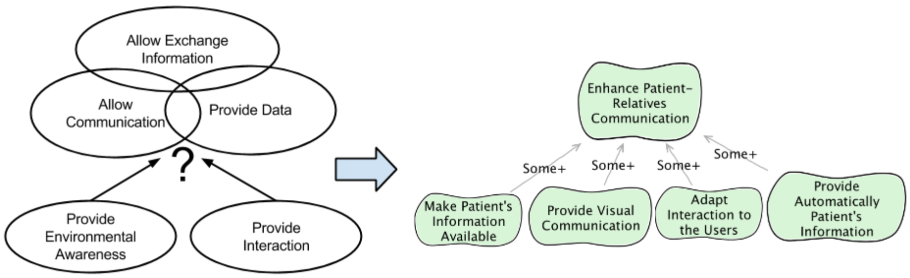
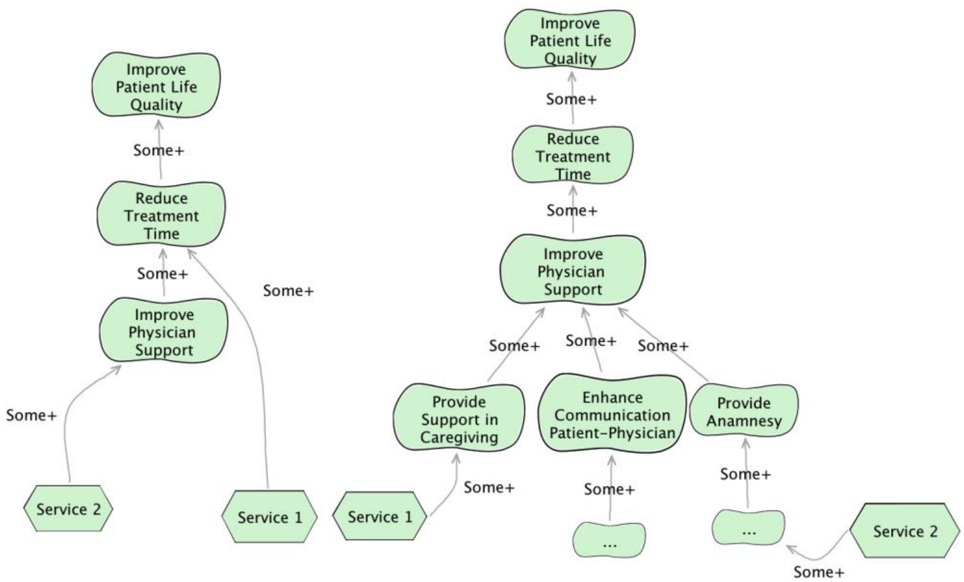


Figure 3. Mapping overlap needs



(a) (b)

Figure 4. Adjusting contribution links

As we are aiming at domain exploration and not at an analysis of a specific solution, it is essential that the evaluation based on the contribution link be as solid and objective as possible. As mentioned before quantifying contribution link is a challenge due to the various viewpoints. However, the way these contribution links were determined can shed light on the actual contribution. We suggest that each of the contribution links be assigned with one of the following:

- There are empirical results supporting the link
- There is analytical evidence provided to support the link,
- Only abstract information is provided, or
- No evidence is provided.

We believe that having an organized and unified graph can shed light over the existing alternatives in a domain, and facilitate the identification of missing parts within the systems implemented within the domain as some of the needs will not be addressed.

3. Concluding Remarks and Future Plans

In light of the research objective – to map a domain requirements and solutions, only few alternatives exist. Among these, one which is more prominent is the *Software Product Line (SPL)* approach which aims at specifying families of products [3]. However, families refer to a specific fragment of a domain. In order to reach integration among such fragments the *Multiple Software Product Lines (*MPLs) notion has emerged. A major means for managing SPL is a *Feature-Oriented Domain Analysis* (FODA) [4] in which features representing the desirable solution can be selected for a specific system. However, this approach neglect the needs view point. In this paper, we adopt a Goal-Oriented Requirement Engineering approach (GORE) and in particular, the i* viewpoint for the purpose of exploring domains in terms of needs and solutions adherence. We were able to map out the needs of the various stakeholders and the relationship among these, as well as the features and goals achieved or provided by various systems. Differently from other approaches the developed domain map is implementation independent. Nevertheless, we encountered a few challenges: it was sometimes difficult to extract and explicit the ambiguous information provided from the various sources; specifying complex interactions among tasks or solutions is not explicitly supported, and managing the abstraction levels to is subject to the modeller intuition.

In the future, we aim at improving upon the required expressiveness and analysis, as well as providing guidelines for performing goal-oriented domain exploration. In particular, we intend to explore techniques that would support the activities within the various stages.

References

1. Sernani, P., Claudi, A., Palazzo L., Dolcini, G., Dragoni, A. F.: Home Care Expert Systems for Ambient Assisted Living : A Multi-Agent Approach, Workshop on The Challenge of Ageing Society: Technological Roles and Opportunities for Artificial Intelligence (2013)
2. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J.: Social Modeling for Requirements Engineering, MIT Press (2011)
3. Pohl, K., Böckle, G., and van der Linden, F. J.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag New York, Inc., USA (2005)
4. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E. and Peterson, A.S.: Feature-Oriented Domain Analysis (FODA) Feasibility Study, Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, (1990)