Revealing the obvious?
A retrospective artefact analysis for an Ambient Assisted-Living project

Itzel Morales-Ramirez, Matthieu Vergne, Mirko Morandini, Luca Sabatucci
Anna Perini, Angelo Susi
Fondazione Bruno Kessler – ICT
38123 Trento, Italy
{*imramirez, vergne, morandini, sabatucci, perini, susi*}*@fbk.eu*

***Abstract*—A variety of methods and techniques for requirements elicitation and analysis have been proposed, in response to the diverse needs posed by the different types of information that have to be managed in designing complex software systems. Experience from real projects gives evidence that often these techniques are combined within a project, but which requirements each technique can better contribute to specify, and which information sources are prevalently used during requirements elicitation and validation is poorly documented.**

In this paper, we describe a retrospective analysis of the requirements engineering phase of a project in the domain of ambient assisted-living, were several elicitation techniques were used to elicit the requirements of a socio-technical system. By empirically analysing the available project documentation, we give a concrete example of the type of information that various elicitation techniques can give in a real project, linking initial sources of information to final requirements through different elicitation paths.

We illustrate the design of this study and present an analysis of the collected data.

***Keywords*-Requirements Engineering, Retrospective case study analysis**

I. Introduction

Requirements engineering (RE) literature reports about a variety of methods and techniques for requirements elicitation and analysis. They can be characterized along different criteria, such as the type of (domain and system) knowledge representation they are good for, and project's complexity dimensions, including stakeholder heterogeneity.

These criteria provide a basis for selecting the most appropriate requirements engineering approach for a project, a problem that has been previously addressed in RE research, e.g. in [12], [9]. From a practical viewpoint, in complex projects different elicitation and analysis techniques are combined, in order to better exploit the different sources of domain information and to model the various perspectives of the stakeholders. This is particularly evident when developing present day socio-technical systems (STSs), systems in which human and technological aspects are strongly interrelated. In fact, eliciting and analyzing the requirements for such complex systems means understanding the involved human organisations in terms of the stakeholders' goals, intentions and resources, and of the role of the technology towards enabling the achievement and maintenance of those goals.

Although detailed account of requirements elicitation and analysis techniques can be found in RE literature, e.g. [4], and project experiences provide evidence that these techniques are successfully combined within a real project, studies about which requirements each technique can better contribute to specify, and which information sources are prevalently used during requirements elicitation and validation, seem missing.

In an attempt to answer these questions, we apply an empirical study, specifically to retrospective case study analysis [15]. Our aim is to investigate whether it is possible to derive empirical data about which knowledge elicitation strategy guided domain analysis and requirements collection, which information sources supported requirements analysis, and in summary from where and how the system requirements were derived.

We study the *ACube* (Ambient Aware Assistance) project[1], which aimed at the development of an adaptive and high quality monitoring infrastructure for an Assisted-Living residence, called *Social Residence*, for elderly people suffering Alzheimer's disease, thus realising a highly developed smart environment as support to medical and assistance staff. The *ACube* system exploits low energy consumption wireless networks of sensors and actuators, which are distributed in the environment, e.g. microphones, cameras and alarms, or embedded in patients' clothes, e.g. biological sensors for ECG. A system level control loop continuously monitors sensor data, analyses them and triggers actuators or specific alarms when critical situations are detected, calling for human operator intervention.

In [13], we sketched a design for a retrospective analysis of requirements artefacts of the *ACube* project, and discussed first findings from the analysis of the data collected on a subset of the project artefacts. In this preliminary study we focused on Goal-Oriented (GO) requirement models used in the project. GO approaches in requirements engineering [17]

[1]Detailed information about the *ACube* project can be found at http://acube.fbk.eu/. The project has been funded by the Autonomous Province of Trento, Italy (2008–2011).

provide concepts and techniques to model social actors, their goals and mutual dependencies for goal achievement. They have been applied in real projects in the health care domain such as [2] and [1], confirming their usefulness to understand such complex domains and to elicit the requirements for STSs in these domains.

In *ACube*, GO modeling was combined with User Centred Design techniques (UCD) [3], mainly because of the complexity of the project domain, with respect to the heterogeneity of the stakeholders of the social residences, which include patients and their relatives, social workers, managers and nurses. Results from the preliminary study motivated us to revisit and consolidate the design of the retrospective analysis of the project, and to execute it on the whole set of requirements artefacts, in light of the specific requirements engineering process used in the *ACube* project.

The rest of the paper is organised as follows. In Section II, we discuss related work. In Section III, we give an overview of the requirements engineering process that was adopted in the *ACube* project. In Section IV, we illustrate the design of the proposed empirical analysis with the possible measures. We present and discuss results extracted from the available documentation, in Section V, and conclude with Section VI.

II. Related Work

When starting a new project, selecting the appropriate RE techniques, in particular for requirements elicitation and analysis, is a key task towards getting good requirements specifications.

The comprehensive survey review on empirical research in requirements elicitation by Dieste et al. [4] derives some conclusions on the relative usefulness of different elicitation techniques, e.g. structured interviews gather more information than unstructured interviews; unstructured interviews gather more information than sorting and ranking techniques; and interviewing is cited as the most popular requirements elicitation method. In [12], a framework is presented, which aims at assisting requirement engineers to perform the selection of the adequate technique, on the basis of six factors, including purpose of requirements, and knowledge types. However, the authors of this work argue that several elicitation techniques have to be combined for capturing requirements in complex software systems, such as STS, in a satisfactorily manner. An analogous framework is proposed in [16], which points out the communication dimension as a key criterion against which to perform method selection. In [8], a characterisation of elicitation techniques is proposed, exploiting an ontology of situational characteristics, which can help analysts in understanding similarities and differences between techniques and in selecting one according to the project domain characteristics.

The above mentioned works give useful hints on specific RE techniques but are less useful to characterise the effectiveness in combining different RE techniques in a given project. For this reason, in our work we are considering a retrospective analysis of a project in which different RE techniques have been integrated, with the aim to get empirical evidence about the effectiveness of their combination, rather than on each individual technique.

For defining our study, we referred to empirical software engineering literature, and specifically to works on postmortem analysis. A review made by Dingsøyr [5] discusses three methods for conducting postmortem analysis, with the objective to "create a culture that promotes continuous learning and fosters the exchange of experience" in an organisation. This work highlights that the central element in knowledge transfer is the dialogue and discussion for discovering what aspects of the process could be improved. Recommendations are given basing on the difficulties found. In practice, during retrospective meetings the objectives and deliverables of the project are revisited together with the process followed in the project. Differently, in our study, we will focus on available project artefacts and involve project analysts only to clarify doubts and to get a deeper understanding of the requirements engineering process adopted. In this respect, our approach is more similar to the one adopted in [11], where a retrospective study devoted to defect analysis is realized by exploiting available project artefacts that represent sources of knowledge (in this case the root causes) and code defects, respectively.

For the design of our study we follow empirical software engineering guidelines, as reported in [15], [6], [18].

III. Requirements Engineering in *ACube*

In the *ACube* project, two main aspects influenced the definition of the requirements engineering process: first, the need of managing the trade-off between cost containment and improvement of quality of services in a specialised centre for people with severe motor or cognitive impairments, such as the involved social residences; second, the fact that the project consortium had a multidisciplinary nature, due to the involvement of software engineers, sociologists and analysts. Moreover, social residence professionals representing end-users were directly engaged in the project's design activities.

The joint use of both UCD [3] and GO approaches [7] allowed to manage the multidisciplinary knowledge between stakeholders by balancing their needs and technical constraints, and in parallel by ensuring the validity, completeness and traceability of requirements. By contract, the requirements analysis phase of the project had a strict deadline of six month after which the technological team needed to receive the requirements to start the system development.

A. RE Methods and Techniques

In *ACube* we exploited the *Tropos*goal-oriented methodology [7], which relies on a set of concepts, such as actors, goals, plans, resources, and dependencies to formally represent the knowledge about a domain and the system requirements. An actor represents an entity that has strategic goals and intentionality within the system or the organisational setting. Goals represent states of affairs an actor wants to achieve. A plan (also activity) is a means to realise a goal. Actors may depend on other actors to attain some goals or resources or for having plans executed. Tropos distinguishes five phases in the software development process: Early Requirements, where the organisational domain is described, Late Requirements, where the system-to-be is introduced in the organisation, Architectural Design, Detailed Design and Implementation.

The elicitation in the field was mainly carried on via UCD techniques. This design approach assures early focus on users, tasks and environment, active involvement of users in the design process. UCD exploits a series of well-defined methods and techniques (from social sciences and psychology) for analysis, design, and evaluation technologies (such as contextual inquiries and scenarios). Techniques used are the analysis of the existing documentation, brainstorming, structured and unstructured interviews, direct observation, questionnaires, goal modelling, and scenario analysis.

In *ACube*, the major sources of domain knowledge were the interviews with the domain stakeholders (in particular, operators, doctors and managers), and the domain document, in this specific case the *Carta dei Servizi* (CS) that describes the services the social residence is committed to give to the patients and to their families and the major activities to be performed to set them up.

Scenarios are stories about people carrying out activities and are used as an instrument to describe instances of behaviour of the system, but their use ranges for several purposes and it is aimed at very different concerns. We distinguish two types of scenarios used at different stages of the process: *activity scenarios* describe a context in which stakeholders act with the aim of summarising, clarifying and reasoning on the collected information, while *technological scenarios* envision the behaviour of a given technology in the context of the project.

B. RE Process and Artefacts

The process followed during the project is detailed in [10], while Figure 1 depicts a view on the process in terms of the main artefacts, used and produced, in process steps. First step in the process is domain understanding, which started with the analysis of the *Sources*. These sources, domain documents and interviews were analysed via *goal oriented techniques*. In this step data coming from the domain is shared across the team and becomes knowledge. So, actors, goals and activities of the organisation are retrieved. Moreover, *criticalities*, which are important events in the organisation, were discovered both from the analysis of the sources via UCD techniques and the Tropos goals and tasks. Such *criticality* can be identified in domain documents and interviews, by considering breakdowns in users' job and procedures or in the use of artefacts. This criticality is documented by putting together narrative description with goals of the model. Each criticality represents a view of the organisation model that focuses on highlighting users, goals and activities to face with a relevant situation, where the system can help to the organisation.

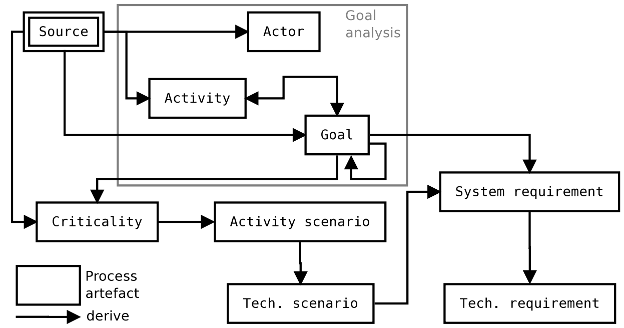

Figure 1. A sketch of the *ACube* requirements engineering process artefacts. Arrows links artefacts that represent respectively input and output in a process step (iterations for refinements not displayed).

The major results of the elicitation and analysis phase have been the definition of a set of *activity and technological scenarios*. In particular, *ACube* system might provide four different activity scenarios: (i) "localisation and tracking of the patients and operators in the residence", (ii) "identification of the behaviour of the patients", (iii) "coordination of caregivers activity with a (semi) automatic report system", and (iv) "therapy management and administering".

Out of these scenarios and of the Tropos diagrams a set of *system and technical requirements*, functional and non-functional requirements, has been generated. A validation session was organised with some of the stakeholders, including 3 managers and 8 operators of nursing homes previously involved in the early exploration phase. The goal of these sessions was the assessment of the validity, acceptability and feasibility of the requirements.

IV. Empirical Study Design

We perform a retrospective artefact analysis of the *ACube* project, retrieving and analysing the available project documentation, which describes the requirements artefacts discussed in the previous section.

To carry out this analysis in a structured and generalizable way we design an empirical study, which is based on the following three research questions:

- RQ1. How did the different information sources contribute to the identification and modelling of the diverse artefact captured in early-requirements documentation (i.e. goal models and criticalities descriptions)?
- RQ2. In which ways did the information sources and the early-requirements artefacts (elements of the goal

models, scenarios) contribute to the elicitation of system requirements?
- RQ3. Does the requirements elicitation process (artefact view), as reconstructed from the empirical analysis of the available documentation, comply with the theoretical process envisaged for the project?

Measures. These questions will be answered applying metrics obtained by performing an analysis on all the available documentation. First, we retrieve the early-requirements artefacts (e.g. actor, goals and plans in goal models and scenarios) used in the project and the traceability links between them (and the documentation that contain their description), which were recorded by the project analysts. These artefacts and the relationships among them, which can be deduced by the traceability links, can then be visualized in a diagram that gives a qualitative reconstruction of the artefact dependencies within the process.

For answering to $RQ1$ we perform a quantitative analysis on the documentation by counting the available elements in the early-requirements artefacts and the respective sources, identified in the documentation. Second, from the available information we try to rebuild a Tropos early-requirements model, with its actors, hierarchies and dependencies, to get a more detailed view on the goals and tasks of each actor in the organisation.

For the second research question, we complete the previous analysis, with the available links going from the early-requirements elements to the requirements. The objective now is to identify and study the different paths recorded during the project that go from the initial sources to the final requirements. By analysing and counting all the links, we can infer to what extent the requirements are related to the actual early-requirements elements and, transitively, to the initial sources.

To answer to the third research question, $RQ3$, we take the information retrieved from $RQ1$ and $RQ2$, with a particular attention to the available traceability links between the various artefacts. We compare the elicitation process resulting from these data with the theoretical process, described in Section III (Figure 1), and highlight the differences. Moreover, the empirical data retrieved regarding the various requirements analysis artefacts is used to verify the compliance of the theoretical process performed against the one reconstructed from the empirical study.

V. Data and Analysis

The analysis of the available *ACube* documentation is performed following the design described in Section IV.

A. Data Collection

We collect all the available documentation on the project: spreadsheets containing lists of early-requirements elements with their links, transcription of interviews and documents textually and visually describing discovered criticalities and

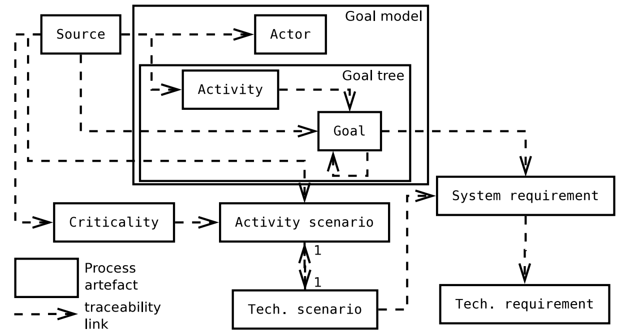

Figure 2. Elements and traceability links identified analysing the available documentation of the *ACube* project.

scenarios. Besides this, we discuss with the analysts for clarifying part of the data and also for refreshing the knowledge acquired during the experience of the elicitation phase.

B. Analysis

The analysis starts with the processing of the tables contained in the spreadsheets by putting together all the links between the sources along with early RE elements — e.g. actors, plans (*activities*), goals, and criticalities— and the final requirements. With the union of these elements we elaborated the diagram in Figure 2. It is worth to be mentioned that criticalities led to the elaboration of scenarios, including both activity and technical scenarios, whereas goals and technical scenarios were the motivation of the system requirements. Besides this, the diagram shows a 1-to-1 relationship between activity scenarios, which deal with the organisation as it is, and technical scenarios, where the new system is introduced. This relationship was not explicitly stated in the traces of the project, but an in-depth reading of the traceability links shows us they are dealing with the same purpose.

Concerning the initial information sources (i.e. domain document and interviews), the interviews produced the larger amount of elements for the early-requirements goal model, as shown in Table I. Looking deeper at the results, we notice that the domain document was the major source of information for discovering the activities to be performed. This finding can be explained considering the fact that the activities represent services that are offered by the actors of the residence to the patients and to the external actors, such as families and control authorities.

Regarding the actors, only few of them are explicitly described in the domain document, since a social residence has the freedom of establishing by itself several roles in the organisation, and just a few roles are fixed at governmental or institutional level. Looking at the single interviews, most actors are added by the first two (held with the coordinators of the structure), which seems reasonable because these stakeholders know the organisational structure at best. In

Information source / Early RE elements	actors	tasks (activities)	resources	goals	criticalities	**sum**
Domain Document *Carta dei Servizi*	5	24	3	3	0	35
Interviews	18	15	18	10	15	76
Tropos Early Requirements Model	0	0	0	12	0	12
Elements found using more than one source	3	12	2	1	0	18
Total number of elements used in the Tropos model	20	27	19	24	/	90

Table I
CONTRIBUTION OF INFORMATION SOURCES FOR MODELLING THE TROPOS ELEMENTS.

contrast, most interviewed partners mentioned resources needed for their work, thus this resources were added to the model quite uniform throughout the interviews.

The goals were retrieved from various sources, in particular from the interviews with the coordinators. However, also a specialised worker, the physiotherapist, gave rise to nearly 15% of the goals, while the social workers did not directly help to reveal new goals. Twelve goals are retrieved indirectly because these goals do not come directly from the sources, but they emerged during the goal analysis phase (i.e. in the Tropos Early Requirements Model), thus the analysts saw the need for adding new goals. Finally, the 15 criticalities are identified through the interviews. This was the choice of the analysts for using interviews instead of the domain document. These criticalities were helpful for illustrating some scenarios of the project. Five scenarios were elaborated which cover from 1 to 3 criticalities. Some criticalities were not covered by any scenario because they were not considered by the analysts for the aim of the project.

Once the first analysis for the identification and counting of early RE elements is completed, we proceed to try to rebuild the early requirements goal model. We create the goal model including all the elements, an excerpt of the goal model is presented in Figure 3. Analysing this excerpt of the Tropos model internal to actor 3, we can identify that interview 2 and 6 contribute for the identification of 2 root goals, whereas during the ER analysis 1 root goal is identified along with its 3 internal goals. Meanwhile, interview 1 contribute with 1 task, both interviews 3 and 6 help for the identification of 1 task, as well as *Carta dei Servizi* together with interview 4 produce 1 task. On the other hand, *Carta dei Servizi* is the source of 2 tasks of the actor 3. From the completed Tropos model we confirm that out of the 12 goals which emerged only during the analysis, 7 are internal goals added to create links between tasks and high-level goals, and the 5 remaining goals are introduced bottom-up, as motivation for an activity. The internal goals are often tacit knowledge [12], which seems either too *obvious* or too *abstract* to the stakeholders, and has thus often to be added by the analysts during goal modelling.

Additionally, the Information Retrieval tool Lucene[2] is used at the end of the reconstruction for searching key terms, (e.g. some words taken from goals' labels), in the digital documents and it gives us hints for the confirmation of some elements emerged during the analysis. The goal model let us state how the various layers are built exploiting different information sources. However we are not able to reconstruct the link between the goal model and the criticalities.

[2]http://lucene.apache.org/

In this way we can answer to RQ1 by showing the quantitative relationship between the information sources and the identified elements. The number of elements gives us a clear reference of the contribution of each information source, whereas rebuilding the goal model help us to understand where the early RE elements come from.

For answering to the second research question we consider the rebuilt early-requirements goal model, and the list of 80 requirements (of which 57 are functional) as the output of the *ACube* requirements elicitation process (Figure 1[3]). The rebuilt elicitation process, shown in Figure 2, was studied, tracking all the paths describing the information flow between sources and requirements. Because of similarities (in the sense they give no additional information), some paths were merged during the study. In particular, we refer to *activity scenarios* and *technical scenarios* indifferently as *scenarios* because of their 1-to-1 relationship. Another simplification comes from the fact that, in the context of the *ACube* project, scenarios are strictly based on criticalities, and no specific information is inferred by considering the different path going through scenarios (with and without criticalities), separately. With these points highlighted, we identify 3 relevant paths between the sources and the requirements, passing through:

- goals
- activities and goals (going through the reconstructed *goal subtrees*)
- criticalities and scenarios

The first path highlights the relation given directly by the goals themselves. For instance, if we look at Figure 3, a source, i.e. *ER*, linked to the goal G14 is also linked to the requirements which come from G14. The second path is not simply the one displayed by the arrows in Figure 2

[3]Notice that for this study we are not considering the technological requirements, which are also part of the output of *ACube* and provide a detailed specification of the technology chosen for realising the system.

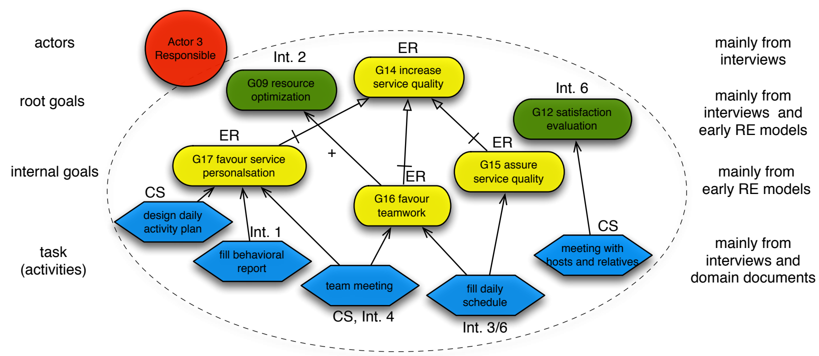

Figure 3. Excerpt of a Tropos diagram representing a nursing home, with an explanation of the various goal model elements and the associated major sources of information in *ACube*. Sources: *Int.* interviews, *CS Carta dei Servizi*, *ER* goals identified during the early RE analysis.

because, in a Tropos model, goals are potentially linked to other goals (subgoals) before to be linked to activities. For our study, we followed the path from sources to requirements through the subelements of the identified goals, aggregating all their sources. For instance, in Figure 3, the goals G15, G16, G17 and the activities linked to them are called the *goal subtree* of G14. For our second path, a source linked to one of the elements of this *goal subtree* is also linked to the requirements coming from G14. The third path is simply the combination of the one-shot arrows through the scenarios and the two-shots arrows through criticalities and scenarios (cf. Figure 2).

Table II, shows some relevant source-requirements links we got by studying the rebuilt elicitation process and Tropos model[4]. The headers of the columns identify the interviews (Int. 1-8), the domain document and the Early Requirement refinements (Early Req.). Following the goals path, we can see that requirements are mainly retrieved from only 2 interviews: 1 (aka the coordinator, whose it is the job) and 6 (aka the physiotherapist, because of a relevant remark which applies in several contexts). Then, they are hugely completed (around 40%) by Early Requirement refinements, all the other documents being almost unused (domain document included). At the opposite, if we look at goal subtree elements, requirements are mainly inferred from the domain document, before to be confirmed (and completed) by *all* the interviews.

However, whatever we look through goals or goal subtrees, the requirements are generally the same in both cases, giving a good way to confirm but not to complete. But there is something to notice in the case of the *ACube* project: a single remark coming from the physiotherapist interview ("act promptly in critical situations") is at the centre of a lot of links because of its relevance in several contexts. This position gives a high importance to the interview but it should be mitigated. Considering this point, a complementary analysis was done removing the physiotherapist interview. In this configuration, we can see some differences passing through goals or goal subtrees: some requirements are identified only from the domain document and the physiotherapist interview, but the domain document is not really used in the goal path. Thus, removing the interview, the requirements identified through the goals only path are significantly decreased. In this configuration, the goal subtree path appears as more complete than the goal path, because 8 more requirements are identified via the domain document. This can be simply explained by the exhaustivity of domain documents, describing details people do not think about during interviews.

Concretely, the remark of the physiotherapist, although it was given for a specific context, was generalised to other situations. This effect can be observed mainly due to the very condensed description of goals in a goal model and the missing (graphical) link to the information sources. Thus, these goals will be perceived by the analysts from a more abstract, high level viewpoint, and decomposed and operationalised accordingly. Moreover, the reliability of the available traceability links was not verified and could thus be a serious threat to validity for the whole analysis. However, for the remaining analysis, the full dataset is used.

Comparing the Tropos model paths (aggregation of goals and goal subtrees) and the scenarios path, we can see a complementarity in the traces: 8 requirements are traced back only through the scenarios and 12 requirements only through the goals paths.

For answering to our third research question, we compare the followed elicitation process, as it was described by the analysts (Section III and Figure 1) to the reengineered process that results from the empirical data retrieved from the project documentation, which resulted in the dependencies illustrated in Figure 2.

The sources for goal model entities seem to be very well documented (although, the documentation did not include goal models that corresponded to the data in the spread-

[4]For a complete description of the links, a detailed table is available at http://selab.fbk.eu/vergne/EmpiRE12/source-reqs-links.pdf

Path \ Source	Interview 1	Int. 2	Int. 3	Int. 4	Int. 5	Int. 6	Int. 7	Int. 8	*Carta dei Servizi*	Early Req.
through goals	16	1	0	0	0	11	0	0	2	22
through goal subtrees	23	5	26	26	18	26	5	11	33	0
through scenarios	28	16	28	0	0	18	19	31	0	0
Sum	67	22	54	26	18	55	24	42	35	22

Table II
AMOUNT OF REQUIREMENTS TO WHICH EACH SOURCE IS RELATED TO, THROUGH THE DIFFERENT PATHS. MOST OF THE 80 REQUIREMENTS IDENTIFIED HAVE MORE THAN ONE SOURCES.

sheets), even if, lacking a history, it is difficult to understand where the goals came from, which define the first version of goal model as their source. Conversely, criticalities and scenarios do apparently not refer directly to the goal model, see Figure 1. Also with a textual analysis of the criticalities we were not able to reconstruct them objectively. Not all identified crititicalities were covered by the so-called activity scenarios. The technical scenarios, where the system-to-be is introduced, refer 1-to-1 to these activity scenarios. They are a main source for the requirements of the system. Moreover, the single requirements refer to goals as their "motivation". Notice, that these goals are modelled in the early-requirements model and thus describe the organisation as it is – i.e. the novel system is not yet introduced. A proper late requirements goal model, including the system actor, could only be reconstructed by inferring goals and actors from the textual description of the system requirements.

As a result, we can state that the well-structured and detailed documentation of the *ACube* RE process facilitated the detailed analysis and interpretation of the data in this retrospective study, and the resulting reconstructed data flow mostly resembled the process which was claimed to be followed by the requirements analysts, although some links were missing. The representation of the data in spreadsheets can thus be recommended to other projects of similar size, which do not use professional requirements tracking tools.

C. Discussion and Threats to validity

The analysis of the project took advantage of a good process documentation which included tables with detailed traceability links. The mix of top-down and bottom-up elicitation confirms the method proposed in [7], was found to be appropriate for this project, but also lead to some incongruence in the multiple paths that can be traced from the requirements back to the information source.

Thus the documentation should be improved for next projects. To ease the reconstruction of the effective path from sources to requirements, it would be of benefit to record the history of models and spreadsheet entries, to capture the sequence of the sources analysed and decisions (addition, modification, deletion) made. Requirements tracking tools offer such functionalities for textual representations. Moreover, a challenge is to build and keep updated early- and late requirements (goal) models (and keeping them updated) would improve the traceability between requirements and motivating goals and help to understand the decisions made when introducing the new system into the organisation.

Regarding threats to validity, the literature in empirical software engineering [18] suggests an analysis of construct, internal, and external validity. According to [14], we analyse construct validity along the following three aspects: intentional, representation and observation validity. The first aspect concerns the selection of the constructs analysed in the study with respect to the objective of the study. In our case the objective was to investigate relationships between requirements engineering techniques and requirements. The constructs analysed were input and output artefacts of the requirements engineering techniques adopted in the project and the resulting requirements specification, thus confirming intentional validity. As for representation validity, the artefact analysis exploited traceability links among elements of the artefacts themselves that were recorded during the project. Observation was focusing on these links, and the quantitative measurements performed are defined in term of counting items, following these links. It is worth mentioning that a validation of the reliability of the recorded traceability links has been performed on a sample of artefacts, using the Lucene IR tool.

Internal validity threats may be identified looking at possible confounding factors related to participants, researchers, and instruments of the study. In our case the artefact analysis has been performed by the authors that were not involved directly in the *ACube* project. The two authors directly involved in the *ACube* project were interviewed to clarify some steps of the process. The measurements were performed manually.

External validity concerns the generalizability of the results, which for case study analysis led to controversial discussion (we refer here to [6]: "Misunderstanding 2: One cannot generalise on the basis of an individual case; therefore, the case study cannot contribute to scientific development." We believe that the *ACube* project can be considered a paradigmatic case of STS. In this sense, the results can be of interest for this type of system. Concerning the provided artefact analysis design, we believe that it can be applied to other case studies, provided the set of requirements engineering techniques adopted in the project, and input and output artifacts to these techniques are available.

VI. Conclusion

In this paper we described a retrospective artefact analysis of a project devoted to the development of an STS for a social residence for people suffering from Alzheimer's disease. The empirical study consisted in performing quantitative and qualitative analyses of available requirements documentation of the project, along three main research questions.

To answer to these questions, we reconstructed the early requirements model and artefacts that were described in the requirements engineering methodology followed for this project. Visualising the links found between the various artefacts, we were able to identify different paths from the requirements through scenarios and goal model back to the original information sources. Finally, we compared the theoretical requirements engineering process followed by the analysts with the process reconstructed from the documentation.

The study was not only made to get an insight to the project and to reveal the information sources for each requirement, but also to examine the success of the elicitation method followed and finally to reveal the importance of a proper documentation of the requirements elicitation phase. Results were not completely obvious. In fact, having been able to reconstruct an important part of the elicitation process, we can state that it mostly followed the envisaged method described in Section III, the mixed top-down and bottom-up elicitation strategy which combined goal modelling and scenario-based elicitation was found to be appropriate for this project, but also lead to some incongruence in the multiple paths that can be traced from the requirements back to the information source. However, the documentation lacked of a version history and the dating of each element added. This would have facilitated the understanding of the information flows and the discovering of inconsistencies in the documentation.

References

[1] R. Ali, F. Dalpiaz, and P. Giorgini. A goal-based framework for contextual requirements modeling and analysis. *Requirements Engineering*, 15(4):439–458, 2010.

[2] Y. An, P. W. Dalrymple, M. Rogers, P. Gerrity, J. Horkoff, and E. S. K. Yu. Collaborative social modeling for designing a patient wellness tracking system in a nurse-managed health care center. In *DESRIST*, 2009.

[3] A. Cooper, R. Reimann, and D. Cronin. *About face 3: the essentials of interaction design*. Wiley India Pvt. Ltd., 2007.

[4] O. Dieste and N. J. Juzgado. Systematic review and aggregation of empirical studies on elicitation techniques. *IEEE Trans. Software Eng.*, 37(2):283–304, 2011.

[5] T. Dingsøyr. Postmortem reviews: purpose and approaches in software engineering. *Information & Software Technology*, 47(5):293–303, 2005.

[6] B. Flyvbjerg. *Qualitative Research Practice*, chapter Five Misunderstandings about Case-Study Research, pages 390–404. London and Thousand Oaks, CA, eds. C. Seale, G. Gobo, J. F. Gubrium and D. Silverman, 2007.

[7] P. Giorgini, J. Mylopoulos, A. Perini, and A. Susi. The Tropos Methodology and Software Development Environment. In Yu, Giorgini, Maiden, and Mylopoulos, editors, *Social Modeling for Requirements Engineering*, pages 405–423. MIT Press, 2010.

[8] A. M. Hickey and A. M. Davis. A tale of two ontologies: The basis for systems analysis technique selection. In *AMCIS*, page 386. Association for Information Systems, 2003.

[9] A. M. Hickey and A. M. Davis. A unified model of requirements elicitation. *J. of Management Information Systems*, 20(4):65–84, 2004.

[10] C. Leonardi, L. Sabatucci, A. Susi, and M. Zancanaro. Design as intercultural dialogue: Coupling human-centered design with requirement engineering methods. In *INTERACT (3)*, volume 6948 of *LNCS*, pages 485–502. Springer, 2011.

[11] M. Leszak, D. E. Perry, and D. Stoll. Classification and evaluation of defects in a project retrospective. *Journal of Systems and Software*, 61(3):173–187, 2002.

[12] N. A. M. Maiden and G. Rugg. ACRE: Selecting Methods For Requirements Acquisition. *Software Engineering Journal*, 11(3):183–192, 1996.

[13] I. Morales-Ramirez, M. Vergne, M. Morandini, A. P. Luca Sabatucci, and A. Susi. Where did the requirements come from? A retrospective case study. 2012. To appear in RIGIM'12 Workshop Proceedings.

[14] D. E. Perry, S. E. Sim, and S. M. Easterbrook. Case studies for software engineers. In L. J. Osterweil, H. D. Rombach, and M. L. Soffa, editors, *ICSE*, pages 1045–1046. ACM, 2006.

[15] F. Shull, J. Singer, and D. I. K. Sjoberg, editors. *Guide to Advanced Empirical Software Engineering*. Springer, 2008.

[16] T. Tuunanen. A New Perspective on Requirements Elicitation Methods. *The Journal of Information Technology Theory and Application (JITTA)*, 5(3):45—72, 2003.

[17] A. van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *5th IEEE International Symposium on Requirements Engineering*, page 249, 2001.

[18] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering - An Introduction*. Kluwer Academic Publishers, 2000.