Design as intercultural dialogue: coupling human-centered design with requirement engineering methods

Chiara Leonardi, Luca Sabatucci, Angelo Susi, Massimo Zancanaro

Fondazione Bruno Kessler (FBK-irst), via Sommarive 18,
38123 Povo, TN, Italy
{cleonardi, sabatucci, susi, zancana}@fbk.eu

Abstract. In the design of information technologies, the challenge of integrating a human-centered design approach with software engineering methods emerge in different forms. The main challenge is to set the ground for different disciplines and professional cultures communicate and work together. The orchestration of different contributions and the establishment of communication practices that facilitates the integration of the different languages and procedures are crucial steps to take full advantage of different research traditions. This paper presents a case study in which human-centered design and requirement engineering methodologies have been used within a large research projects aiming at developing innovative technologies and services to support professionals in nursing homes. The design process took the form of an intercultural dialogue that required human-centered and requirement-engineering professionals to work across borders. Starting from our case study, the paper presents the boundaries identified between the human-centered and the engineering perspective and proposes a framework to guide the integration process conceived as an intercultural dialogue between disciplines.

Keywords: Human-centered design, Requirement engineering, inter-disciplinarity

1 Introduction

In recent years both human-centered and requirement engineering communities are facing the challenge of identifying ways to combine and integrate their respective approaches in order to cope with the growing complexity of systems and infrastructures to be designed. From one side, a human-centered approach facilitates the design of technologies that best fit users practices and therefore are less likely to be rejected; on the other side, requirement engineering approaches ensure an efficient and cost-effective way of designing these technologies.

The need of combining the two approaches is particularly crucial within the field of ubiquitous, context-aware computing because for this kind of technologies interaction of technical, organizational, social and ethical issues is posing serious challenges to requirement gathering and user-assessment of the technological solutions proposed [1]. Furthermore, all those aspects should be tackled looking at the technology from two levels, namely, the level of services and the architectural level [2].

The issue of requirement gathering and management in this context is crucial and Lyytinen et al. [2] well synthesize the challenge:

"requirements capture is not just how to identify and incorporate technical requirements, but how to foresee and integrate social and organizational requirements with those capabilities. These are often hard to identify before actual deployment of the ubiquitous computing service. In order to successfully design and implement ubiquitous computing, actors who were separated in time and space in the development cycle need to establish much tighter coordination patterns. The development methods and processes for ubiquitous computing environments will need to reflect such new needs".

Both requirement engineering and human-centered design address this challenge from different points of view and the opportunities for cooperation are widely acknowledged.

Yet, several issues remain unexplored and a number of questions are still open.

The two fields have their own established methods and practices to organize knowledge and to define and communicate requirements as well as different vocabularies. Moreover, they entail different views concerning the validity of methods and hence concerning the final results of the research.

The issue of how to combine these two approaches has been recently discussed by several authors. Kaindl [3] poses the issue concerning the usefulness of common representations for requirements engineering and interaction design and the possibility and the convenience in developing a combined process. Nebe et al. [4] also argue that a fruitful cooperation is possible if we first identify integration points between disciplines.

In this paper we try to address some of the challenges posed by a joint use of the two approaches within the same design process. In particular, we discuss a case study in which a "dialogic" relationship between the disciplines was sought that allows researchers to cooperate and share knowledge taking advantages of the reciprocal strengths of the two approaches.

Our case study investigated the design of smart environment in nursing homes at support to the medical and assistance staff as part of a larger research project focused on intelligent environments. The ACube system employs a network of sensors distributed and embedded in the environment or embedded in the users' clothes to allow an efficient and unobtrusive monitoring of the nursing homes guests. The ACube project team was multi-disciplinary and included software engineers, sociologists and interaction designers. Representative end users were also directly engaged in design activities.

For the design phase, two teams were involved, one with experience with the qualitative-oriented methodologies of interaction design (in particular the use of personas and scenarios) and one with experience in the use of the semi-formal approach to requirements engineering called Tropos, a goal-oriented software engineering methodology that allows to describe the domain and the stakeholders' needs and to reason on concepts such as actors, goals and social relationships [20]. The initial attempt to define a common vocabulary for requirements engineering and

human-centered design teams was the source for philosophical and methodological discussions that eventually culminated with the feeling that something different should be tried.

That confrontation was not useless: the team reached the awareness that the true problem is, first of all, epistemological and requires a serious consideration and reflection on the assumptions behind the two different approaches. It was first of all clear that a combination of the two approaches was necessary but this combination should have not to be sought in terms of reducing either one approach to the other. Our strategy, thus, was articulated around three main points:

1. to identify strengths and limits with respect to the specific design objective, in order to define integration points between human-centered and Goal-oriented requirement engineering methods and take full advantages of their reciprocal strengths;
2. to identify main barriers that make the dialogue difficult and inefficient;
3. to define communication protocols and tools to make the dialogue effective.

The design process took the form of an intercultural dialogue that required human-centered and requirements engineering professionals to work across borders.
Starting from our case study, the paper presents the boundaries identified between the human-centered and the engineering perspective and proposes a framework to guide the integration process conceived as an intercultural dialogue between disciplines.

2   Defining the relationship between disciplines

Several experiences are concerning the integration Requirement Engineering and human-centered design techniques. Sutcliffe and Maiden [5] propose a framework for relating scenarios to use cases: scenarios are automatically generated from use cases and are validated by rule-based frames able to detect problematic event patterns. Lamsweerde [6] and Rolland [7] claim a bi-directional relationship between goals and scenarios: goals lead to the elaboration of scenarios and scenarios may prompt the elicitation of goals or may be used for goal inference. Uchitel and Chatley [8] combined the use of goal oriented requirement elicitation and scenario and discussed some techniques for validation aims. Cockburn [9] suggests the use of goals to structure use cases by connecting every action in a scenario to a goal of another use case at a lower level of abstraction. In this sense a scenario is built each time a goal is discovered. Yu and Liu [10] propose a technique for supporting reasoning on social models by means of scenarios. They use scenarios expressed in Use Case Maps to describe elaborated business processes or workflow. Aoyama [11] uses both scenarios and provisional personas in order to identify a set of requirements for a web application in which the kind of users is wide and difficult to classify.

The problem of integrating methods coming from the human-centered and requirement engineering research areas without losing the strengths of the two approaches can be tackled, in our view, once we recognize that the main challenge is to make the setting for a dialogue between disciplines. The first step is hence the understanding of the relationships and boundaries between the different approaches

and the opportunities for combining them. Traditionally, two main trends can be identified for composing research approaches of different nature: from one hand, there is the tendency to develop overarching and unifying frameworks [12], on the other hand the effort to make different disciplines speaking to each and collaborate across borders.

2.1 Limits of unifying frameworks

In the case of unifying framework, one disciplinary approach is usually modified to be assimilated into the other approach, or several approaches are modified to fit into an overarching meta-model. While the risk is to limit the potential of the approach itself; the advantage is to work in a situation of 'methodological purity' where a certain epistemology has been chosen to drive the process.

In software engineering, for instance, the Situational Method Engineering [13, 12] is grounded on the assimilation approach: constructing ad-hoc software engineering processes by reusing fragments of existing design processes; the basis for the assimilation technique is the method fragment [14], a self contained component that can be used as building blocks for the process composition. Techniques for fragment manipulation (extraction, selection, and composition) are still open points, and even if there is a disagreement about the level of precision, it is clear that fragment specification requires a language for describing at least the process and its products. Some recent approaches [15] make use of the SPEM notation[1] for describing the process as a workflow and meta-models as linguistic keys for bridging activities and artifacts coming from different methodologies [16]. In particular the meta-model approach demonstrated working fine in the context in which the integration concerns two engineering approaches that are based on the same epistemological nature. In the field of human-computer interaction also there have been several attempts to integrate the different disciplines and approaches composing the fields (activity theory, ethnography, distributed cognition, etc.). As stated by Rogers [17]:

"a problem with integrating such different theories and ontologies, however, is that it becomes difficult to know what frames of reference and axioms to use for a given problem space. (...) It seems that only the researchers who have developed the grand theories are able to use them".

Rogers argues that the attempt to develop overarching frameworks has the advantage of break away from the confines of specific disciplines. On the other side, the risk is to have inefficient and difficult to apply framework that can suffer from the "toothbrush" syndrome, that is relevant for the one that developed the theory but, unfortunately, for no one else [17].

---

[1] SPEM – Software & System Process Engineering Metamodel Specification, Version 2.0 available online at http://www.omg.org/spec/SPEM/2.0/

2.2 Toward a "dialogic" perspective

By "dialogic", we mean a perspective of interaction that is grounded on communication and iterative confrontation. Differently from what happen with unifying frameworks, this perspective recognizes the irreducible cultural difference between the two approaches and therefore asks practitioners to work in a situation of epistemological and methodological pluralism: the goal being that not to transform or to assimilate an approach to make it fits into another one, but rather to bridge the gap between different research traditions to take advantages of their mutual strengths. In this case we have a dialogic process, where various approaches coexist and where each approach can hold more relevance depending on the specific problem to be addressed and on the phase of the research.
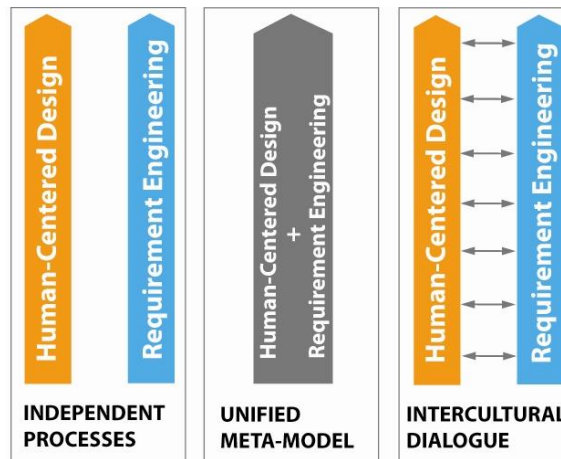
Figure 1 – Possible models of relationships between Human-Centered Design and Requirement Engineering

A dialogic perspective, therefore, does not require transforming and adapting the methods specific to each research tradition but requires that preconditions for a beneficial dialogue between the two are defined and set. The challenge here is to manage the dialectic issue concerning the concurrent usage of different research paradigms: an engineering approach such as Tropos and less formalized and qualitative approach such as the human-centered ones.

Even if the approach seems promising, it raises several issues. In particular, the different epistemological and methodological traditions have to be first understood by practitioners of both disciplines in order to be integrated.

Studies on cross-disciplinary and interdisciplinary research are helpful to understand the challenges of the collaboration between disciplines. Eigenbrode et al. [18] argue that the main challenges of cross-disciplinary research are related to the linguistic and conceptual divides and to the perceived nature of the world: researchers may have different views on what is reality and how reality can be known (e.g. constructivism vs realism). The authors posit that these are mainly philosophical challenges related to conflicting assumptions and propose a toolbox for philosophical dialogue, consisting

of a set of questions designed to encourage confrontation on scientists'' philosophical views on their research.

Strober [19] also illustrates how the collaborative process is neither easy nor intuitive. Fundamental barriers are not only linguistic, but cultural. While translations are pretty an easy task, what is much more difficult is to understand and accept assumptions, methods and ways of validating the truth employed in other disciplines.

In the following, we present from our case study, the process that emerged from the confrontation of the two research teams, the challenges that rise during the process and finally propose some lessons learned from this experience.

3 Tropos: a Goal-Oriented Requirement Engineering method

For the sake of clarity, we introduce here the Tropos method, a semi-formal methodology [20] that 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 organizational setting. An actor is used to model both human stakeholders and software and hardware systems. Goals represent states of affairs an actor wants to achieve. Executing a plan can be a means to realize a goal. Actors may depend on other actors to attain some goals or resources or for having plans executed (see Figure 3 for an example). Tropos models are visualized through actor and goal diagrams. The former are graphs whose nodes represent actors and arcs are strategic dependencies between pairs of actors. A goal diagram represents an individual actor perspective in terms of its main goals, and their decomposition into sub-goals. Furthermore, plans and resources that provide means for goal achievement are depicted through means-end relationships.

The semi-formal nature of the Goal-Oriented languages gives, in general, an effective way to communicate with the technical staff and carries some other advantages. In particular: it allows for a structured representation of the requirements; it allows to specify automatic and semi-automatic conflict analysis procedures to refine the set of requirements and to highlight, as soon as possible, requirements inconsistencies; finally, it allows to establish a clear traceability between the motivations for the existence of a given requirement and the requirement itself by linking it to the domain needs described via the goal models.

Tropos distinguishes five phases in the software development process: Early Requirements, where the organizational domain is described, Late Requirements, where the future system is introduced in the organization, System Architecture Design, System Design and System Implementation.

The use of the Tropos methodology enables structuring the domain knowledge in terms of actors, goals social dependencies and other concepts that are sharable with the UCD team. Moreover, the five phases process allows a systematic tracing of the domain knowledge towards the system implementation artifacts, with the result of establishing a bridge between requirement elicitation (in collaboration with UCD techniques) and system deployment. Specifically, in the ACube project, we applied the first two phases of the Tropos methodology, in order to sketch the strategical

dependencies among the actors participating to the elderly setting, and to anticipate the impact of introducing the ACube system into the organization.

4   Case study: The design of ACube

ACube[2] is a large research project aiming at designing a highly technological environment to be deployed in nursing homes as a support to medical and assistance staff. The project envisages a network of sensors distributed in the environment or embedded in users' clothes that should allow monitoring the nursing home guests without influencing their usual daily life. Advanced automatic reasoning algorithms allow acquiring the data through the sensor network and to promptly recognize emergency situations or prevent possible dangers for the guests themselves. An important activity in the project was the analysis of the system requirements with the objective of having a quality improvement of services in the nursing home.

The whole requirements elicitation and analysis process within ACube was articulated in seven phases all of them characterized by the use of both the approaches (see Figure 2).

Phase 1 – Field data collection. The process started with the investigation of the domain to understand the organizational setting in four nursing homes and to identify the needs of the involved stakeholders. First, users are classified in: primary users, which are direct users of the product (e.g., caregivers); secondary users, which are those who will occasionally use the artifact or those who use it through an intermediary (e.g., managers); and tertiary users, which are persons who will be affected by the artifact or influence its success (e.g., patients, family members). Several human-centered design methods exist in order to get rich insights about the context to be explored. Recently contextual inquiry [21] demonstrated the capacity to satisfy the needs for a deep and, at the same time, quick understanding of complex domain. Contextual inquiry mainly consists in interviewing people in their context, preferably when performing their tasks. Contextual inquiries were performed in 4 different nursing homes and involved about 40 health professionals.

---

[2] ACube is funded by the local government of the Autonomous Province of Trento in Italy.
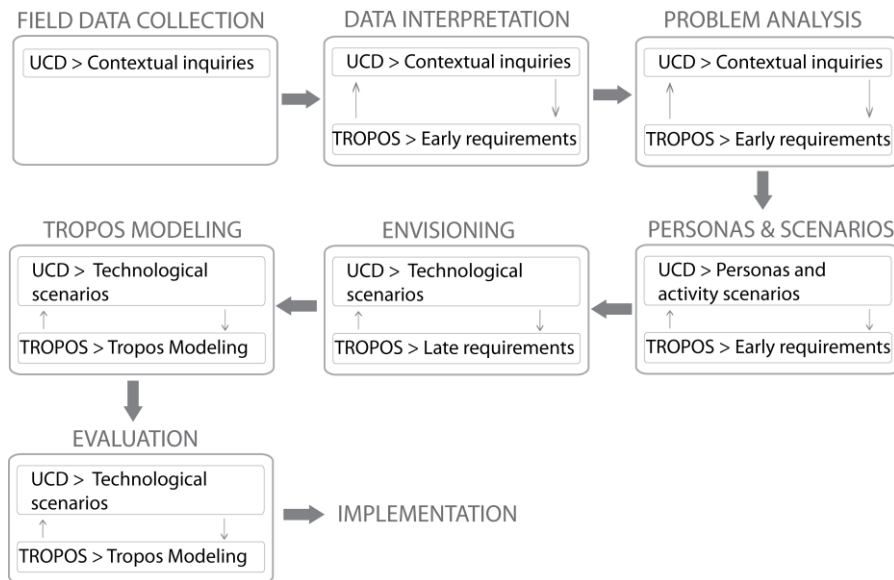
Figure 2 - The project whole cycle

Phase 2 – Data Interpretation. The data interpretation is the step in which data coming from the domain is shared across the team and becomes knowledge. In our process, data interpretation is concurrently carried out in a twofold way: i) following contextual design approach dimension (flow model, sequence model, cultural model, artifact model, physical model) [21] and ii) exploiting the early identification phase of Tropos. The Early Requirement process is used for modeling the initial set of domain entities. It includes the Actor diagram and the Goal diagram. The Actor Diagram provides a bird-eye view over the domain, by specifying how responsibilities are distributed across roles, and providing an intuition of which interactions occur in the environment. An actor diagram is a graph whose nodes represent actors (agents, positions, or roles), while edges represent dependencies among them.

Phase 3 - Problem analysis. The analysis of critical aspects was developed to highlight main problems that professionals of nursing homes experience in their job. The Criticality Identification is an activity that bridges the Tropos analysis with the following *personas* and scenario authoring. The aim is to highlight every possible breakdown or problem that may occur in the organization that hinders the achievement of goals, such as the situation related to the intervention for a fall of one of the patients. A criticality connects an exceptional event to front with Tropos goals and tasks that are identified to receive a negative contribution. It also encapsulates the context in which the problem may occur. A criticality could be identified in the documents produced by contextual inquiry, by considering breakdowns in users' job and procedures or in the use of artifacts, and it is documented by putting together the narrative description with goals of the model. Each criticality represents a view over

the organization model that focuses on highlighting users, goals and activities when an emergent problem occurs. Criticality identification triggers the subsequent consolidation and envisioning steps, by providing motivations for the introduction of a system into the domain. Not all entries get the same importance, thus a prioritization is important for reducing the risk to focus on marginal problems of the domain, or infrequent situations. Critical aspects that may be addresses through a technological intervention, such as the problem related to the fall, were represented through Tropos Early Requirements diagrams and led to activity scenarios and personas (in which technology and services were not yet present).

Phase 4 – Personas and activity scenarios design. We adopted the specific scenarios approach as developed by Rosson and Carroll [22] and subsequently enriched by Copper with the notion of *personas* [23], Personas are rich descriptions of archetype users meant to draw attention on users' goals and motivations [24]. Introducing *personas* in scenarios-based approach provides an anchor against self-referentiality in design and make scenarios more concrete. A set of activity scenario-descriptive scenarios focusing on problems and criticalities identified through user studies – were developed. The starting point for the Phase 5 is constituted by 5 personas and 5 activity scenarios.

Phase 5 – Envisioning, from data to design. The phase of envisioning moved from personas and activity scenarios identified in Phase 4 in order to envision how to introduce the technology in the analyzed domain.

A participative workshop was organized to identify how the system could support the critical situations identified in Phase 2 and hence develop technological scenarios to be further evaluated with end-users representatives and stakeholders. About 10 participants attended the workshop including the ID team and representative of stakeholders and technologists. The heterogeneity of the group was meant to guarantee the generation of creative but feasible ideas, to provide concrete solutions to problems identified by nursing homes professionals as well as to provide solutions that could meet engineers' expectations and their research interests. Outcomes were pursued at multiple levels: to expand the designer's prospective and to watch the problems from different points of view, to figure out how their ideas can work in a real context, to identify design criticalities and open issues, to generate requirements of the system-to-be. The workshop ended with the definition of 5 different macro-services the ACube system might provide.

As a consequence of the envisioning focus group, and the introduction of the system into the organization, the Tropos process moved from the early requirement phase to the late requirement phase. Figure 3 shows an excerpt of the Tropos model, describing a small part of the goals and the activities of the *SeniorOSS* actor (a caregiver in the nursing home). In particular the actor *Senior OSS* has the goal to [`avoid dangerous behavior of patients`] that can be AND decomposed in [`monitor patients in her visual area`] and [`coordinate interventions in the nursing home area`].
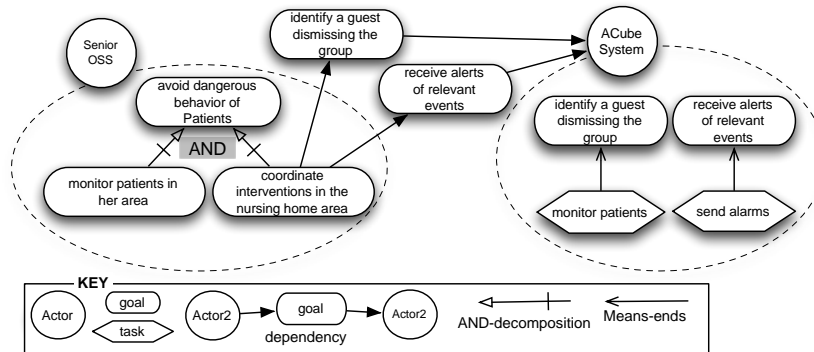
Figure 3 - An excerpt of the Tropos model for the nursing home

Phase 6 - From design ideas to Tropos modeling. Tropos diagrams and scenarios were jointly used to refine the ideas emerged during the creative workshops. On one hand, technological scenarios were designed to make design ideas concrete and to trigger reflection about possible services. On the other hand, Tropos diagrams were developed to more systematically analyze how the introduction of a system impacts on the domain actors. In particular, Tropos allows for describing the relationships between the stakeholders and the system, represented as a new domain actor, in terms of actor dependencies.

This way, the envisaged services provided by the system are described within their operational environment, along with relevant functions and qualities. Modeling dependencies between actors and the system introduced in the organizational setting provides a more rigorous view on the envisioned services and helps designers in understanding critical dependencies between actors and goals. Despite scenarios and *personas* are very effective tools to show particular instance of a problem and to figure out design solution, they do not provide adequate support for abstract thinking. Tropos, on the contrary, provides an exhaustive view on the domain that is crucial to go back from specific design ideas to abstract requirements, to consider alternatives when designing novel services, and to balance the impact that each design decision have on non-functional requirements of the system. In addition the formalization of requirements supports automatic and semi-automatic procedures for the verification of completeness and conflict analysis.

Phase 7 – Evaluation of technological scenarios. Here visual scenarios were derived from the Tropos models and used for the validation phase, where multiple views on the domain are required to drive the negotiation and refinement of requirements with stakeholders and project partners. If envisioning scenarios provided a concrete instance of a particular design solution, that is very helpful to discuss with stakeholders, on the other hand, Tropos diagrams aided designers in reconsidering design solutions and elaborating alternatives thanks to the possibility to trace back design solution to initial abstract requirements.
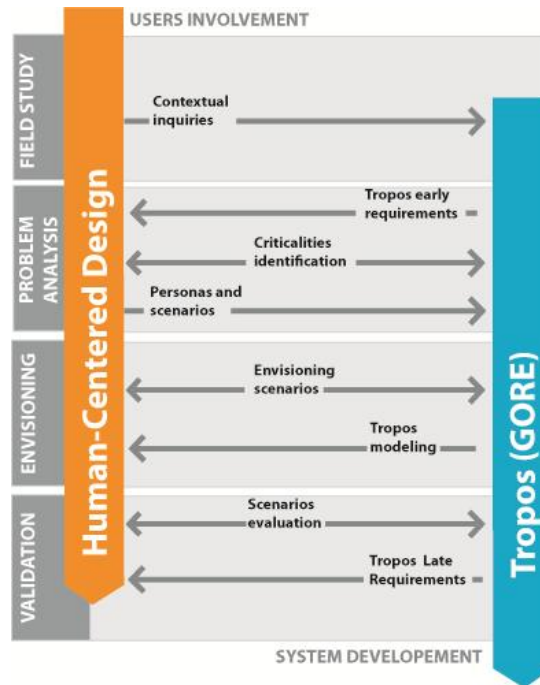
Figure 4 - The relationships between HCD and RE for our case study

A first focus group was held with the representatives of the 10 research groups involved in ACube project, 27 people attended the meeting. The second focus group was organized with the stakeholders, 3 managers of nursing homes previously involved in the early exploration phase attended the meeting. The goal of these meetings was to assess of the validity, acceptability and feasibility of requirements and to envision alternatives not considered in the scenarios.

The workshop with technological partners was focused on technical feasibility and research interest, and on the envisioning of original solutions to the critical situations identified. Acceptability and usefulness were instead the pivots of the workshop with end-users representatives. Expected outcomes of the two focus groups were the emergence of design criticalities, the resolution of open problems and the identification of new propositions and ideas, in order to collect additional elements to elaborate an organic description of the technological architecture.

The output of this phase was the agreement on early requirements and the refinement of Tropos late requirement diagrams in order to choose the main functionalities of the system, such as the need of having ACube monitoring critical events such as the fall of the patients (see Figure 5).

Figure 5 - An example of scenario used for the validation phase

In the following sections we reflect on the role of the two approaches and identify the boundaries existing between them.

5 Reflections on the experience

After the design experience, we could assess our initial strategy based on the three main mechanisms that we employed to shape the relationships and set the dialogue for an efficient cooperation between the two teams (see section 1):

- *Strengths/limits analysis*: identify strengths and limits of the methods while achieving a given design objective in order to define integration points between HCD and RE methods and take full advantages of their reciprocal strengths;
- *Barriers identification and exploration*: identify main barriers that make the dialogue inefficient;
- *Mutual learning*: enable continuous information exchange and communication to overcome the barriers coming from the inter-cultural and multi-perspective approaches to facilitate the negotiation of the meanings of terms and the sharing of common modeling tools during the requirements elicitation and analysis activities.

In the following, we recap and discuss both what we realized during the actual work as well some new and unexpected issues in an attempt to reason about and abstract our experience to highlight problems and proposing guidelines.

4.1 Exploring reciprocal strengths and limits

The first step requires identifying strengths and limits in order to define integration points between HCD and RE methods and take full advantages of their reciprocal strengths. This analysis was beneficial for the team because promoted mutual learning between practitioners and was a starting point for identifying integration points as a basis to define the research process and the methods to be exploited in the different phases of the process. The output of this analysis is synthesized in Table 1. The awareness of the differences is a first step to further define and overcome existing divides between approaches.

Table 1 - Strengths/limits analysis

Tropos Modeling	Human-Centered Design
Abstract representation of the domain	Concrete representation of the domain
Semi-formal representation of requirements	Informal/narrative representation of the requirements
Static and invariant picture of the domain	Dynamic representation involving the spatio-temporal dimension
Do not provide specific tools for finer prioritizing requirements than the reasoning on alternatives and contributions	Scenarios provide a support for prioritizing requirements
Neutral representation that do not engender an emotional response	Personas-based scenarios are sort of "dramatic" representation that engenders empathy
Do not provide information about the physical context	Provide details about the physical context in which people act
Provides a general representation of invariant dependencies among actors	Provide details about how interactions occur in a given specific situation
Support requirement traceability	Do not support traceability
Support automatic requirements conflict-analysis and check of quality properties	Do not support conflict-analysis
Provide tools to efficiently communicate and validate requirements with technical staff	Provide tools to efficiently communicate and validate requirements with end-users and non-technical partners

4.2 Making the divides explicit

Once reciprocal strengths and limits of the two approaches were identified, an exploration of the barriers that can prevent a synergy between the two approaches was pursued. As studies in the field of cross-disciplinary research suggest, an effective collaboration between disciplines entails that differences are explored and assumptions made explicit. We identify two main challenges that researchers should face in order to establish a dialogue: the epistemological and conceptual barrier.

Epistemological divides. The first main issue is to consider epistemological foundations and validity criteria of both the approaches, to manage differences without weakening and distorting the two research paradigms. While requirement engineering – and hence Tropos - is grounded on a positivist research tradition [25], several methods employed in HCD origin within a constructivist and interpretative tradition (ethnography, ethnomethodology, etc.).

Positivism is an epistemological perspective that claims the knowledge is based on sense experience and positive verification. One of the key features of positivism is the ability of demonstrating the logical structure and coherence of a concern by axiomatization. Tropos is classified as a positivist approach (even if the debate on the positivist nature of many RE methods has recently been criticized [26]) by providing a precise frame for the modeling activity and the reasoning process. For instance, typical instruments for modeling the domain are: (i) the abstraction is used to reduce the complexity of the representation of the domain instance to the eyes of an observer, by focusing only on specific domain properties that are considered central to the analysis; (ii) decomposition that is used to break the complexity of an entity thus to separately study its components when moving from generic towards specific; (iii) refinement is used to derive specifications, by using argumentation and logic deduction, based on the analysis of existing data.

Constructivism recognizes that our knowledge is socially constructed and it does not reflect an external "transcendent" reality. Knowledge is hence contextual and contingent on convention and social experience.

The exploration of the epistemological divide of the two research tradition is crucial of we consider, for instance, the two different perspectives on the nature of requirements: while in RE tradition requirements often "exist out there" and the role of the analyst is to capture them, manage them, and validate them; on the other side, within the HCD approach, requirements are the output of a negotiation that happens during the whole design life-cycle [27]. Requirements are constructions produced by a number of actors (users, analysts, developers, designers) each acting in specific context and each having specific motivation and perspectives on the system to be designed.

Linguistic and conceptual divides. RE and HCD approaches exploits very specialized dictionaries and concepts. Often, the same terms (e.g. the concepts of 'role', the concepts of 'requirement') have very different connotations and entails different methodological choices. One of the more discussed divides concerns the view and the consequent treatment of requirements [27]. If we define a requirement as something to be captured or extracted from the users certain methods should be preferred, on the other hand, if we define requirements as something to be collaboratively constructed other methodological choices may be preferred. The conceptual definition in this case has important consequences on the whole design cycle.

A linguistic boundary is due to a mismatching in the dictionary used in the two methodologies. This aspect is specifically evident in the integration between a semi-formal language (Tropos) and an informal, often verbose and intentionally ambiguous language. The identification of these linguistic boundaries is important for the reconciliation of incompatible concepts and for creating the framework for knowledge sharing.

An example of linguistic boundary is the Tropos 'task' and the human-centered 'activity' terms. A Tropos task is defined as the conceptualization of a plan that provides the means for the operationalization of a goal. An example of task is [caregivers monitor guests' behavior]. The human-centered design activity concept

captures additional information about the context in which it is carried out, including the user point of view and empathy:

"... during my job it is important to continuously observe patients' behavior, but this is often an heavy activity to carry on together with other our duties. This is due to the high number of guests compared to the low number of professionals. This working overhead causes we are incapable of concentrating on the human aspect of our job as well as we would do ... " (extracted from an interview with a caregiver).

It is clear that it contains some cues (for example the frustration about the difficulty of giving priority to the human aspect) that are difficult to encode in a plan-based representation but indeed represent important information for the design.

Another example is represented by the concept of 'persona' with respect to Tropos' actor. Whereas both of them identify users of the future system, an actor is a way to abstract a role in the organization, while a persona is an archetype of user, sufficiently concrete to provide the understanding within a scenario. The cognitive and emotional dimensions are important factors which persona tries to catch for helping the designer to take decisions in the design process, characteristics that are missing in an actor.
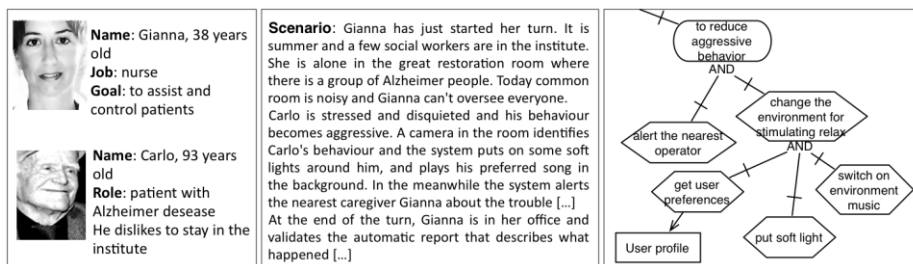
Figure 6 - examples of artifacts: personas, technological scenario and Tropos late requirement diagram

A similar problem emerged with the definition of 'role'. Within Tropos, a role is defined as an abstract entity that is a holder of responsibilities within the domain, owns strategic goals and may execute plans to achieve them. In Tropos roles may be bounded by dependencies, relationships that establish *depender*, *dependee* and *dependum*.

For human-centered researchers, role is a more fluid and ambiguous concept, encompassing formal and informal responsibilities. For instance, tacit responsibilities of caregivers working within a nursing home are those related to the human aspect of their work, as for instance, communicate emotional closeness to guests.

4.3 Mutual learning

The exploration of strengths and limits as well as the analysis of the divides are in our view the prerequisites for a dialogue between the two approaches.

Mutual learning represents the crucial aspect to mediate between the different epistemologies and languages hold by requirement engineering and human-centered

design. Mutual learning is here intended as something happening through action. Knowledge, following Piaget [28] can only be gained in acting upon the real and transforming it in order to understand it. This view of knowledge and mutual learning poses several challenges to research teams that begin to work together.

The first issue is to explore how Tropos and human-centered concepts are related. In our experience the mutual learning process has been achieved through the (i) definition of a shared dictionary of terms that includes all the most important concepts employed by the two methodologies; (ii) engagement in a collaborative negotiation of the definition of terms in the dictionary which leads to discover hidden relationships between terms; and (iii) the iteration and the refinement of definition until the agreement is acceptable.

The shared dictionary of terms is an important instrument for activating the communications among people with different background and professional skills. We selected the natural language as the most appropriate language to use since its expressiveness and flexibility is well suited for explaining and hence sharing the meaning of terms coming from both the methodologies. The shared dictionary has been a reference point for enabling an easily and autonomous translation among the internal knowledge of each team, coded according to a specific knowledge representation format.

The collaborative definition of terms in the dictionary represents the instrument for engaging in a mutual learning process. This is specifically useful for the appropriation of the terminology used in the other disciplines. We can say that the appropriation of the other discipline's terminology happens when a member of the team defines, using his own terminology, a terms coming from another discipline, that may produce uncertainty or different interpretation. The iterative revision of the dictionary leads to refine the definitions in order to reach a better precision. The effort to refine definition by using other terms of the dictionary may be rewarded by discovering hidden relationships between terms.

6   Conclusions

Both human-centered and requirement engineering approaches, in particular Tropos, ground their processes in gathering knowledge about the people and their activities. In both cases the goal is to provide knowledge to design information system. However, they not only have different set of techniques and incompatible vocabularies but also they are based on two diverging epistemological foundations. Human-centered practitioners shun from formal methods and principally aim at providing rich information about the users and the context in which they carry out their activities.

On the other side, requirement engineering approaches are grounded on accountable and formal or semi-formal procedures that are exploited throughout the design life-cycle: from requirements analysis to systematic and complete system description. The "engineering perspective" to system design is well suited for managing requirements within the design life-cycle and in providing guidance for transitioning from informal to (semi-)formal knowledge and for analyzing conflicts between requirements. For instance, Tropos, - a representative of the Goal-Oriented

requirement engineering approaches - plays a fundamental role in the development of enabling reasoning about the domain features with the aim of identifying conflicts and of checking for validity of functional and non-functional requirements.

In this paper we addressed some of the challenges posed by a joint use of the two approaches starting from the experience done within the ACube project. We discussed how a "dialogic" relationship between the disciplines may provide guidance for researchers from requirement engineering and human centered design field that cooperate within the same design process. In this perspective, the orchestration of different contributions, the establishment of communication practices and the engagement within a mutual learning process are presented as crucial steps to take full advantage of different research traditions.

References

1. Abowd, G.D.; Mynatt, E.D.; Rodden, T.; The human experience [of ubiquitous computing]. In Pervasive Computing, IEEE Jan-Mar 2002
2. Lyytinen, K. J.; Yoo, Y.; Varshney, U.; Ackerman, M.; Davis, G.; Avital, M.; Robey, Da.; Sawyer, S.; Sorensen, C. (2004) "Surfing the Next Wave: Design and Implementation Challenges of Ubiquitous Computing," Communications of the Association for Information Systems: Vol. 13, Article 40.
3. Kaindl, H., Constantine, L., Pastor, O., Sutcliffe, A., Zowghi, D. How to Combine Requirements Engineering and Interaction Design?. In Proceedings of the 2008 16th IEEE International Requirements Engineering Conference (RE '08). IEEE Computer Society, Washington, DC, USA, 299-301.
4. Nebe, K.; Zimmermann D. Aspects of integrating user-centered design into software engineering processes, (2007). In Proceedings of the 12th international conference on Human-computer interaction: interaction design and usability (HCI'07), Julie A. Jacko (Ed.). Springer-Verlag, Berlin, Heidelberg, 194-203.
5. Sutcliffe, A.; Maiden, N.; Minocha, S.; Manuel, D. Supporting scenario-based requirements engineering, IEEE Transactions on Software Engineering, vol. 24, no. 12, pp. 1072–1088, 1998.
6. Van Lamsweerde A.; Willemet, L. Inferring declarative requirements specifications from operational scenarios, IEEE Transactions on Software Engineering, vol. 24, no. 12, pp. 1089–1114, 1998.
7. Rolland C., Achour C. B., Cauvet C., Ralyté J., Sutcliffe A., Maiden N., Jarke, M., Haumer P., Pohl, K., Dubois E., Heymans P. 1998. A proposal for a scenario classification framework. Requir. Eng. 3, 1 (Sep. 1998), 23-47.
8. Uchitel S., Chatley R., Kramer J., Magee J.. Goal and scenario validation: a fluent combination, Requir. Eng., 2006, Vol: 11, Pages: 123 - 137, ISSN: 0947-3602.
9. Cockburn, A. Structuring use cases with goals. Journal of Object-Oriented Programming, vol. 10, no. 5, pp. 56–62, 1997
10. Liu L., Yu E.. Designing Information Systems in Social Context: A Goal and Scenario Modelling Approach Information Systems, Volume 29, Number 2, April 2004 , pp. 187-203(17).
11. Aoyama M. Persona-and-Scenario Based Requirements Engineering for Software Embedded in Digital Consumer Products, in Proc. of the 13th IEEE International Conference on Requirements Engineering, 2005, p. 94
12. Henderson-Sellers B.; Ralyte J., Situational Method Engineering: State-of-the- Art Review," Journal of Universal Computer Science, vol. 16, no. 3, pp. 424-478, 2010.

13. Brinkkemper, S. Method engineering: engineering of information systems development methods and tools, Information and Software Technology, vol. 38, no. 4,pp. 275- 280, 1996.
14. Cossentino, M., Gaglio, S., Garro A., Seidita V. Method fragments for agent design methodologies: from standardisation to research," International Journal of Agent-Oriented Software Engineering, vol. 1, no. 1, pp. 91-121, 2007.
15. Seidita, V., Cossentino M., Gaglio S., A repository of fragments for agent systems design," ICAR-CNR, Tech. Rep., 2006.
16. Susi, A.; Perini, A.; Mylopoulos, J.; Giorgini, P. The tropos metamodel and its use," Informatica, vol. 29, no. 4, pp. 401- 408, 2005.
17. Rogers, Y. New Theoretical approaches for Human-Computer Interaction. Annual Review of Information, Science and Technology, (2004) 38, 87-143.
18. Eigenbrode, S.; Wulfhorst, J. D.; Althoff D. M.; Goldberg, C. S.; Merrill, K.; Morse, W.; Nielsen-Pincus, M.; Stephens J.; Winowiecki, L.; Bosque-Pérez, N. A. "Employing Philosophical Dialogue in Collaborative Science," In BioScience, 57, 2007, pp. 55-64.
19. Strober, M. H. Interdisciplinary Conversations: Challenging Habits of Thought. Stanford University Press, 2010.
20. Bresciani, P.; Giorgini, P.; Giunchiglia, F.; Mylopoulos, J.; Perini, A. Tropos: An Agent-Oriented Software Development Methodology. Autonomous Agents and Multi-Agent Systems, 8(3):203–236, 2004.
21. Beyer, H.; Holtzblatt, K. (1998). Contextual Design: Defining Customer-Centered Systems. San Francisco: Morgan Kaufmann.
22. Rosson M. B.; Carroll J. M. Scenario-based design. In The human-computer interaction handbook, Julie A. Jacko and Andrew Sears (Eds.). L. Erlbaum Associates Inc., Hillsdale 2002.
23. Cooper, R. Reimann, and D. Cronin, About face 3: the essentials of interaction design. Wiley India Pvt. Ltd., 2007.
24. Grudin J., Pruitt J.. Personas, participatory design and product development: an infrastructure for engagement. In Proceedings of Participatory Design Conference 2002, June 2002.
25. Weber, R. "The rhetoric of positivism versus interpretivism: A personal view," MIS Quarterly, vol. 28 (1), pp. 3–12, 2004
26. Haumer, P. Pohl, K., Weidenhaupt K. Requirements elicitation and validation with real world scenes, IEEE Transactions on Software Engineering, vol. 24, no. 12, pp. 1036–1054, 1998
27. Robinson M.; Bannon L. Questioning representations. In Proceedings of the second conference on European Conference on Computer-Supported Cooperative Work (ECSCW'91), 1991, Liam Bannon, Mike Robinson, and Kjeld Schmidt (Eds.).
28. Piaget, J. 'Piaget's theory'. In P.H. Mussen (Ed) Manual of Child Psychology, vol.1, 3rd Edition. (1970) London: John Wiley.