Exploring the Boundaries: when Method Fragmentation is not Convenient

Chiara Leonardi[1], Luca Sabatucci[1], Angelo Susi[1], and Massimo Zancanaro[1]

Fondazione Bruno Kessler IRST, Via Sommarive, 18 I-38050 Trento, Italy
{cleonardi,sabatucci,susi,zancana}@fbk.eu

Abstract. This paper presents an approach to explore the coupling of User-Centred Design and Tropos methodologies. The two methodologies have been employed in a real project aiming at developing smart environment for nursing home to support medical and assistance staff. In particular Tropos has been used for modeling (and reason about) the domain and the system, whereas User-Centred Design has been useful for establishing an interface for communicating with stakeholders. The integration was challenging due to the epistemological differences between the two design approaches.

1 Introduction

Goal-oriented Requirements Engineering (GORE) plays a fundamental role in the development of *user intensive systems*, enabling reasoning about the domain features with the aim of identifying conflicts and of checking for validity of functional and non-functional requirements. Nevertheless, we experienced the need for an effective way to center the design on the users of the system. The strength of Goal-Oriented techniques in modelling the domains can be still enhanced by coupling the engineering perspective with a creative perspective typical of User-Centred Design (UCD) approaches. Basic principles of this integration are: (i) early focus on users, tasks and environment, (ii) the active involvement of users in the design process, (iii) allocation of functions between user and system, (iv) the incorporation of user-derived feedback into system design, (v) iterative design whereby a prototype is designed, tested and modified.

Both approaches ground their processes in information about the people that are directly or indirectly involved by the technology that has to be developed. Yet, they not only have different set of techniques and incompatible vocabularies but also they are based on two diverging epistemological foundations. UCD practitioners shun from any formal method at risk of compromising the actual use of the knowledge gained in the field. On the other side, RE practitioners often loose contacts with "real" people because formalizations cannot easily be shared with them: user analysis thus becomes a single-player game rather than a meaningful dialogue with stakeholders.

Context. The need of reconciliating the two approaches raised up from the work in a large research project aimed at developing a smart environment in

nursing home as support to medical and assistance staff[1]. Coming from some experiences in situational method engineering [1, 2] we supposed to use method fragment composition and meta-model unification for creating an ad-hoc design process for our aims. Anyway the attempt to define a common vocabulary for engineers and sociologists team members was the source for long philosophical discussions that terminated with the feeling that other ways should be walked. In particular the main problem was to find an agreement on identifying precise relations between terms coming from different vocabularies. In addition, the situation became more complicated when we tried to understand phases and activities to perform (and work product to produce); whereas Tropos life-cycle is clear and well-defined, UCD practitioners refuse to decide a-priori the activities to perform in a project and their order. Anyway, the attempt was not useless: we reached the awareness that the problem is not only dialectical, but epistemological; the two research teams intend their process and language from different points of view. It was clear that a combination of the two approaches was necessary but that combination should avoid reducing either one approach and the other. We perceived the paradigm should change from fragment composition to fragment collaboration.

Contribution. The contribution of this paper is the analysis of foundations for the integration of the engineering perspective of Tropos with the qualitative prospective of the UCD approach. While goal oriented requirement engineering provides accountable procedures and formal or semi-formal methods for eliciting requirements and providing systematic and complete system description, the UCD process encompasses less formal practices aimed at envisioning sparking ideas by inspirational techniques ranging from ethnographic fieldwork for understanding users, to design storm to inspire 'blue sky' concepts. We propose a framework for mediating these two different approaches without compromising their very nature: in the differences, it lies the power of the integration and its risks. The framework is based on a novel concept for creating a synergy between methodologies (or parts) that is the definition of communication protocols between methods. This concept is based on the exploration of methodological and linguistic boundaries and the definition of channels for sharing and tracking design data among practitioners of the involved methodologies.

2 Challenges

Traditionally, two main trends can be identified for composing research approaches of different nature: from one hand, there is the tendency to privilege a disciplinary perspective and, on the other hand, the effective effort to integrate different epistemologies [3]. In the first case one disciplinary approach is usually modified to be *assimilated* into the other approach: while the risk is to limit the potential of the approach itself; the advantage is to work in a situation of 'methodological purity'. In the case of disciplinary *integration*, practitioners should accept to work in a situation of methodological pluralism: the goal is not

[1] ACube project, funded by the Autonomous Province of Trento. http://acube.fbk.eu/

to transform or to assimilate a specific approach to make it fits into another one, but rather to bridge the gap between different research traditions and take advantages of their mutual strengths.

This distinction can be borrowed from social science to be applied in software engineering. Indeed, the Situational Method Engineering [4, 5] 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 [6], a self contained component that can be used as building block 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 [7] make use of the SPEM notation for describing the process as a workflow and meta-models as linguistic keys for bridging activities and artifacts coming from different methodologies [8].

The preconditions for applying situational method engineering is to analyze core elements of a methodology for building model descriptions of activities and artifacts. There are cases in which these preconditions do not apply, and a formalization of the process is not feasible without the risk to loose all advantages of the process. In our case, whereas Tropos phases and diagrams can be formalized by using meta-models and SPEM diagrams [8, 7], UCD practitioners are very resistant in providing any kinds of structure for framing their theories and techniques; they shun from any formal method at risk of compromising the actual use of the knowledge gained in the field. They claim the freedom is the key for flexibility of procedures and for quickly adapting to the context. In addition, the language they use exploits ambiguity as a design opportunity and not as a problem: the everyday world is inherently ambiguous, and allowing this ambiguity to be reflected in design has the advantage to encourage people to interpret situations, by establishing deeper and personal relations with the meaning offered by situations [9].

For these reasons UCD artifacts have typically a descriptive form that preserves every information about the domain, ranging from users' motivations to the empathy versus the product. Therefore, it is not reasonable (and productive) to generate a meta-model without reducing the expressiveness of these artifacts. In these cases the integration approach may be useful, as a replacement of the assimilation approach; this because it does not require transforming the methods specific to each research tradition but it is based on creating preconditions for a beneficial dialogue between the two [10]. Maintaining the different epistemological and methodological traditions is grounded on managing the dialectic issues concerning the concurrent usage of different research paradigms.

By exploring the boundaries between Tropos and UCD, two sub-challenges have emerged and are discussed in the following.

2.1 Epistemological challenge

The first 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 Tropos is grounded on a *positivist* research tradition [11], several methods employed in UCD derive from a *constructivist* perspective.

Positivism is an epistemological perspective which holds that knowledge is based on sensorial 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 [12] — by providing a precise frame for the modeling activity and the reasoning process.

Constructivism is a different epistemological perspective which holds that the Ontological Reality is utterly incoherent as a concept, since there is no way to verify how one has finally reached a definitive notion of Reality: scientific knowledge is built by scientists and not discovered from the world. In this context, there is no single valid methodology and researchers play an active role in defining the reality. UCD is grounded on this research tradition: hence the scarce formality of methods, the subjective insights developed by practitioners, and the ambiguities in the analysis are, if correctly managed, not only accepted but actively perused [9, 13].

Each methodology has its own basic axioms that not only guide the research process, but also the way method is perceived and applied. In the spirit of integration rather than assimilation, the concrete procedure proposed in this paper does not dictate a choice or a priority among the two approaches but rather leaves analysts free to choose the most promising techniques in the two domains. Indeed, the process boundary to explore and to overcome is peculiar: making explicit and reason on these differences is the first step to exploit the complementary nature of UCD practices and RE approaches.

2.2 Linguistic boundary

Beside the methodological boundary, a linguistic boundary exists. The concurrent usage of both approaches requires that a common language exists in order to make a dialogue possible. Several concepts exist in both Tropos and UCD that suggest an integration is possible and profitable. Examples of these are the pairs of goal/need, actor/persona, task/activity; yet these terms have slight different meanings in the two methodologies that hinder the composition process.

The integration of the two methodologies must pass through a reconciliation of terms. Two alternatives were possible: (i) to create a unified meta-model of the integrated process, or (ii) to tie up terms with similar meanings while keeping them separate. The first way was fascinating but it failed because of the difficulty to identify a meta-model for the UCD process. Just as an example, during the attempt to formalize a term like 'persona', we had the feeling to loose

the flexibility and the expressivity of the instrument. Thereafter, the definition of a communication protocol for allowing an exchange of data between the two processes revealed preferable even if it required an additional effort for creating a framework in which data of different nature can be easily interchanged.

The proposed framework maintains the original nature of the instruments and it introduces new methodological and conceptual tools for tracing all data transformations along the process. In the following, some differences are identified between Tropos and UCD techniques. For example, in our case the field study leads to the identification of a number of institutional roles for our stakeholders. Then, Tropos engineers used these data for defining relevant actors of the domain, their main goals and dependencies. Afterwards, the analysis tried to generalize information in order to discover high level interdependencies among these actors. On the other side, UCD practitioners focused on the behaviour of individual workers, during their daily job, keeping track of attitudes and personal motivations that may influence the final value of the design. From the description of stakeholder daily activities (including routines, methodologies, but also unexpected situations to front) the Tropos engineers extracted goal/task decomposition and resource usage. The UCD designer, again, focused more on problems, stressing situations, lacks of methodologies and expectations over the system under design. These activities were continuously intermingled until the awareness of the knowledge of the domain was considered deep enough.

3 Methodology Integration

The *Tropos methodology* [14] is a goal-oriented design process 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. Goals represent states of affairs an actor wants to achieve. A Plan is a means to realize 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 organizational domain is described, Late Requirements, where the system-to-be is introduced in the organization, System Architecture Design, System Design and System Implementation.

User Centered Design is a design philosophy that exploits a number of different techniques within a iterative design process. Tenets of UCD are: early focus on users, tasks and environment, the active involvement of users in the design process, allocation of functions between user and system, the incorporation of user-derived feedbacks into system design, iterative design by which a prototypes is designed, tested and modified. UCD exploits a series of well-defined methods and techniques coming from social sciences and psychology for analysis, design, and evaluation technologies. Contextual inquiries, personas and scenarios - that we adopted in our project - are widely used when researchers aim at obtaining a rich picture of a context (organizational, social, physical), at easily communicat-

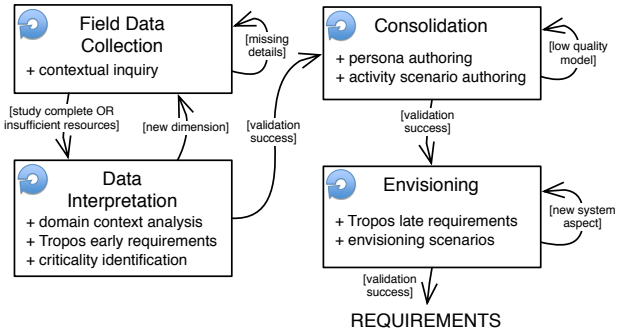

Fig. 1: Phases of the process and life-cycle

ing it to stakeholders in order to envision acceptable and innovative technological solutions.

The framework for the integration of the two methodologies considers both epistemological and language boundaries discussed in the previous section. The result is an integrated methodology where the component processes maintain their own identity, even if their activities are interleaved and an intensive exchange of data is supported by specific communication protocols.

3.1 Integration of lifecycles

The process model is represented in Figure 1, that shows phases, activities and conditions for moving along the steps. Every macro-phase of the process is represented as a box with a title and activities are placed inside. The execution order of activities in a phase is not specified: they are concurrently managed and iterated as well as some conditions are met. Generally, each phase terminates with a validation according to assigned criteria, in which typically also stakeholders are involved.

The process begins with the investigation of the domain in order to understand the organizational setting and to derive possible needs and services that the system could provide to users. Several methods exists to analyse the domain: recently ethnographic methods, such as contextual inquiry, demonstrated their capacity to satisfy the needs for a deep but at the same time rapid understanding of complex domain. Data interpretation provides a first classification and abstraction in order to create a believable model of the domain, but avoiding to loose important details typical of a narrative analysis. In our process, data interpretation is concurrently carried out in a twofold way: one is the domain context analysis and one is the Tropos early identification. The data consolidation acts as a filter in order to focus on relevant characteristics to consider in the following phases. Finally, the envisioning phase lets the analysis team to reason on the system-to-be in order to expand designers' prospective, to look at the problems from different points of view, to figure out how their ideas can work in

a real context, to identify design criticalities, and to generate requirements. The process ends with the validation of requirements with stakeholders, essential for moving to the next design phase.

3.2 Exploring Methodological Boundaries

The modeling activity in the Tropos methodology follows a 'positivist' approach, indeed, Tropos algorithms and meta-model [8] provide well-defined descriptions for the design activity. A typical shortcoming of positivist approaches is that they can not easily provide general techniques for interpreting the domain, and transforming perception data into model elements. In the case of Tropos, for instance, it is the analyst's responsibility to decide how to model the domain, managing trade-offs and choices: the process does not provide general guidelines about what actors and goals to include in the model, how to handle and/or decomposition, and so on.

On the other side UCD is a 'constructivist' approach, thus it avoids prescribing a process to follow, but it provides criteria for achieving project objectives and supporting the designer decisions. For instance, the contextual inquiry phase, in which the designer gathers detailed data needed for the design and discovers implicit aspects of work that would normally be invisible. This activity may be conducted by using different techniques (interviews, direct observation, questionnaires, and so on) to use in isolation or to interleave according to the needs emerging from the context.

Data captured by contextual inquiry is greatly useful for leading decisions during the early requirement; between these two activities there is a methodological boundary that may be used to create a method synergy. The contextual inquiry produces a huge documentation concerning observed users/customers and their needs. If opportunely analyzed and filtered this data can feed the Tropos entity identification, by providing criteria for motivating the introduction of new elements and tracing the source.

Nevertheless, another methodological boundary exists in the opposite direction: filtered data, modeled with Tropos, is an input for UCD designers in order to feed the following consolidation phase. An example is the Tropos early requirement that produces a model of the domain, organization dependencies and stakeholders' strategies for goal commitments. This model can be profitable used by UCD designers in order to summarize relevant aspects of the domain preliminary to the envisioning of the new system.

3.3 Exploring Linguistic Boundaries

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 well-specified language (Tropos) and a language that is intentionally verbose and sometimes ambiguous (UCD). The identification of these linguistic boundaries is important for the reconciliation of incompatible concepts and for creating the framework for data sharing.

An example of linguistic boundary exists between the Tropos 'task' and the UCD '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 UCD 'activity' concept captures additional information about the context in which it is carried out, including the user point of view and the empathy aspect. An instance of activity description is extracted from an interview to a caregiver:

> "... 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 ..."

Maintaining and tracking this difference along the unified process is fundamental for the following design phases, but it requires a reconciliation: it requires to explore how a Tropos task is related to an UCD activity. The solution we explored is to connect the two concepts with a different kind of relationships respect to classical ones used in meta-modeling. We introduce a loose relationship among linguistic elements that is based on collaboration protocols. This is discussed in the following section.

3.4 Defining Collaboration Protocols

In a situation of methodological pluralism, in which at least two design teams collaborate, a collaboration protocol relates linguistic elements that need to be reconciliate. A protocol defines crossing terms, steps, guidelines and instruments for translating and tracking design data from one methodology to the other.

The exploration of methodological boundaries provides hooks in which a collaboration is possible and beneficial, whereas linguistic boundaries identify elements that must be reconciliate in order to realize the collaboration. An instance of collaboration protocol is defined for the boundary existing when moving from the Tropos early identification to the UCD consolidation activity, which — in our project — was conducted with activity scenarios and personas authoring.

The Tropos *Early Requirement* activity depicts the strategic and organizational views of the domain. During this phase the relevant stakeholders are identified, along with their respective objectives; stakeholders are represented as actors, while their objectives are represented as goals. Goal and plan models allow the designer to analyze goals and plans from the perspective of a specific actor. This phase results from the analysis of social and system actors, as well as of their goals and dependencies for goal achievement.

The use of *Scenarios* in RE is pretty established as an instrument to describe instances of behavior of the system, but their use ranges for several purposes and it is aimed at very different concerns [15]. We used activity scenarios, which are stories about people carrying out activities; they describe a context in which personas act with the aim of summarizing, clarifying and reasoning on the

collected information; these scenarios are narrative description of the behavior of personas in critical contexts of the domain [16]. Another difference respect to classical software engineering scenarios is the use of personas. *Personas* are powerful instruments for creating descriptive models of system-to-be users [17]. Mikkelson and Lee [18] introduced user archetypes that describe classes or types of user of a product, further refined by Cooper [17] that introduces "personas" as composite archetypes based on behavioral data gathered from many actual users encountered in ethnographic study. They provide a tangible representation of the user to act as a believable agent in the setting of scenario. Summarizing, personas are hypothetical but significant user archetypes for which to motivate the design; they are defined as [19–22]: (i) attitudes, experiences, aspirations; (ii) general expectations the persona may have about the experience of using the product; (iii) behaviors that persona will expect from the product; (iv) how the persona think about basic elements or units of data.

A linguistic boundary was identified between the Tropos concept of 'actor' and the UCD concept of 'persona'. Whereas both of them identify users of the system-to-be, an actor is a powerful instrument to abstract a role in the organization, while a persona is an archetype of user, sufficiently concrete to provide the understanding of the empathy emerged from ethnographic study and personal motivations within a scenario. The cognitive and emotional dimensions are important factors persona try to catch for helping the designer to take decisions in the design process, characteristics that are missing in an actor.

The collaboration protocol for this couple — methodological/linguistic — of boundaries is based on the identification of criticalities that tie up the organization model with the concrete context in which actors play their roles. The *Criticality Identification* bridges the Tropos early requirement analysis with the following persona and scenario authoring, by connecting linguistic terms like actor and persona.

A criticality is an exceptional situation to front in the organization for which the system is designed. The criticality is discovered in the Tropos goal-models, by analyzing and/or decomposition and by the conflict analysis. A criticality is identified as a view on the organization model that focuses on highlighting actors, goals and tasks when a critical situation occurs. The description is enriched with information about the context in which the problem may occur and the impact on the standard stakeholder activities.

The aim of this protocol is to highlight every possible breakdown or problem that may occur in the organization that hinders the achievement of goals; this information — given to UCD designers — leads the construction of scenarios that, subsequently, have a specific significance for reasoning of system requirements in the creative sessions. Criticalities are initially classified and prioritized on the base of their relevance in the domain. Subsequently for each relevant criticality at least one scenario is authored and a cast of personas is engaged. The aim of the scenario is to highlight concrete instances in which the problem occurs, and to reveal stakeholders' behavior in the circumstance.

4 Discussion

The first point we want to discuss in this paper is whether fragmentation activity is always possible or — as well as in the case of 'constructivist' approaches — it is a risk to reduce the advantages of methodology synergy. The difficulty to frame a design process within a precise formalization may hinder the applicability of situation method engineering techniques.

In our framework we maintained the two epistemologically different methodologies, by creating some communication channels for the teams of engineers and sociologists to easily communicate and share information. This activity required a deep analysis of the two approaches, in order to identify where the two methodologies present similarities and where they where conceptually different. At the beginning, for this purpose, the teams spent time in defining a common vocabulary. During these meetings the participants identified pairs of terms that may be re-conciliated (actor/persona, goal/need, task/activity), but they failed in identifying a precise relation between them even if relations have been investigated. The problem were not only dialectical, but epistemological: the two teams have different sensibility and a different vision about the problem and how to solve it. For instance, an actor identifies the 'abstraction' of a role in an organization, whereas a persona is an 'archetype' of users for which the system is going to be designed; the actor is featured with institutional goals that hold for every person will play the role, whereas each person is unique due to his/her personal attributes. This way the failure in the definition of a unified vocabulary raised up the need for the exploration of the boundary between component methodologies. Boundaries have to be interpreted as an additional value for the integration, because they allow for defining how to share design data even if talking different languages.

The second point of this discussion is the systematization of the approach we exploited. The goal is to reconciliate the use of communication protocols with existing situational method engineering techniques for constructing methodologies. In our opinion it is possible to consider a communication protocol as a specific fragment, built ad-hoc for the specified situation. Considering, for instance, the criticality identification activity used to tie up the Tropos early requirement with personas/scenarios authoring: this activity did not exist neither in Tropos nor in UCD. The concept of critical aspect and the technique for identifying criticalities in the domain have been created ad-hoc for linking Tropos activities/concepts with UCD ones. Now, we are investigating whether a communication protocol can be framed inside the situational method engineering by introducing a loose methodological and linguistic link for integrating fragments. It is worth noting this is a loose relationship, that is different from a 'strong' structural relationship because it does not introduce destructive modifications. For instance, aggregation is the classic meta-model link among elements, and it is typically used when blending two meta-models (or portions) [6]. Another advantage of the 'communication' link is that it does not require a full formalization of the process and the products concerning the fragment. It may work even when the process model is

partially defined or in other cases — as UCD — in which the process frequently changes with the context and the language is flexible but ambiguous.

References

1. M. Cossentino, L. Sabatucci, and V. Seidita, "A collaborative tool for designing and enacting design processes," in *SAC*, S. Y. Shin and S. Ossowski, Eds. ACM, 2009, pp. 715–721.
2. M. Cossentino, L. Sabatucci, V. Seidita, and S. Gaglio, "An agent oriented tool for method engineering," in *EUMAS*, ser. CEUR Workshop Proceedings, B. Dunin-Keplicz, A. Omicini, and J. A. Padget, Eds., vol. 223. CEUR-WS.org, 2006.
3. A. Pickard and P. Dixon, "The applicability of constructivist user studies: how can constructivist inquiry inform service providers and systems designers," *Information Research*, vol. 9, no. 3, pp. 9–3, 2004.
4. S. Brinkkemper, "Method engineering: engineering of information systems development methods and tools," *Information and Software Technology*, vol. 38, no. 4, pp. 275–280, 1996.
5. B. Henderson-Sellers and J. Ralyté, "Situational Method Engineering: State-of-the-Art Review," *Journal of Universal Computer Science*, vol. 16, no. 3, pp. 424–478, 2010.
6. M. Cossentino, S. Gaglio, A. Garro, and V. Seidita, "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.
7. V. Seidita, M. Cossentino, and S. Gaglio, "A Repository of Fragments for Agent Systems Design," in *Proc. Of the Workshop on Objects and Agents (WOA'06)*, Catania, Italy, September 2006.
8. A. Susi, A. Perini, J. Mylopoulos, and P. Giorgini, "The tropos metamodel and its use," *Informatica*, vol. 29, no. 4, pp. 401–408, 2005.
9. W. Gaver, J. Beaver, and S. Benford, "Ambiguity as a resource for design," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM New York, NY, USA, 2003, pp. 233–240.
10. R. Weber, "The rhetoric of positivism versus interpretivism: A personal view," *MIS Quarterly*, vol. 28 (1), pp. 3–12, 2004.
11. C. Potts and W. Newstetter, "Naturalistic inquiry and requirements engineering: reconcilingtheir theoretical foundations," in *Proc. of the Third IEEE International Symposium on Requirements Engineering*, 1997, pp. 118–127.
12. C. Hinds, "The case against a positivist philosophy of requirements engineering," *Requirements Engineering*, vol. 13, no. 4, pp. 315–328, 2008.
13. T. Wolf, J. Rode, J. Sussman, and W. Kellogg, "Dispelling design as the black art of CHI," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, p. 530.
14. L. Penserini, A. Perini, A. Susi, and J. Mylopoulos, "High variability design for software agents: Extending Tropos," *TAAS*, vol. 2, no. 4, 2007.
15. C. Rolland, C. Ben Achour, C. Cauvet, J. Ralyté, A. Sutcliffe, N. Maiden, M. Jarke, P. Haumer, K. Pohl, E. Dubois *et al.*, "A proposal for a scenario classification framework," *Requirements Engineering*, vol. 3, no. 1, pp. 23–47, 1998.
16. P. Wright, "What's in a scenario?" *ACM SIGCHI Bulletin*, vol. 24, no. 4, p. 12, 1992.