Engineering Normative Requirements

Alberto Siena
ITC-Irst, Trento, Italy
e-mail: siena@itc.it

***Abstract*— Requirements engineering is the critical phase of the software engineering process, where complex, heterogeneous human needs, together with environmental constraints have to be captured and translated into system requirements. In this paper we recall the concept of normative environment, discuss the problems and challenges related to the requirements analysis in this kind of environments and present an approach for dealing with such problems. Finally, we apply the approach to a simple case study to verify its suitability and to highlight open points to be addressed in the future.**

***Index Terms*— Software Engineering, Systems Engineering Methodologies, Requirements Analysis.**

I. Introduction

Looking at the real world, we can observe that software-intensive systems have often to deal with regulatory contexts. A “software-intensive system” is the sum of software and hardware, and the human activities that complete the system. With “regulatory context” we mean the layer of a system, which articulates stakeholders’ intentionality with regulations, policies, laws, and so forth. We group all these deontics-related constructs under the word “norms”. A system, which is comprised by both autonomous stakeholders and a regulatory context, is a “normative environment”.

Requirements Engineering (RE) is the discipline that aims at understanding the purpose for a software-intensive system to exist, and its desired functionalities. Inadequate, ambiguous, inconsistent requirements can compromise the quality of the software. In particular, Goal-Oriented Requirements Engineering (GORE) methodologies aim at supporting decisions about requirements elicitation through the analysis and comprehension of the stakeholders’ intentionality (goals). The way they deal with regulatory contexts is still ambiguous; the common approach for including the regulatory context is to view norms as special goals or requirements.

However, these approaches seem to have some drawbacks. First, they delegate a wide range of actions to the responsibility of the software engineer; for instance, operationalize regulations into agents’ behavior, verify agents’ goals reachability under regulations and check agents’ behavior consistency against regulations. Second, there is no support for the possible improvement of the regulatory context, like the possibility to explore alternative regulations that could articulate agents’ behavior in a better way. Finally, if the regulatory context is complex, the corresponding complexity of a goal model can become intractable. The lack of formal support for these actions can cause time losses, risk of wrong elicitation and increasing costs.

In recent years, in the RE community a new growing attention has been paid to the necessity of capturing such aspects into a formal framework, in order to be able to analyze them computationally, and have more rigorous treatment. As far as we know, in the literature there are examples that pertain to security-related topics, privacy and access policies. We believe that these approaches are only *partial views* over a *larger picture*; in our work we aim to identify and capture the larger picture.

The objective of this paper is twofold. On the one hand, it aims at highlighting the need for techniques, which allow software engineers to deal with normative environments, reason about their requirements, and make effective decisions about them. On the other hand, it demonstrates, through a preliminary experiment, where a potentially successful solution could come from, and what are the future work and difficulties.

The paper is structured as follows: section II positions our work in the literature and overviews some useful background; section III introduces the problem and depicts the direction for a possible solution; section IV illustrates a reference scenario, which will be used to explain the addressed problems; section V presents operationally a preliminary approach and identifies some interesting results; section VI concludes, by presenting the challenges that the approach rises and the future work.

II. Background and Related Work

In Multi-Agent Systems, the role of a regulatory layer has always been recognized as an important metaphor in order to achieve objectives such as coordination and collaboration among autonomous agents [10], [7], [1].

In Software Engineering, an important contribution comes from the e-Institution framework [18], [21]. An e-Institution is a software realization of an institution, which in turn is “a set of artificial constraints that articulate agents interaction” [19]. An e-Institution is implemented as a variation of a finite state machine, which defines the illocutions that agents can exchange. However, there is no consideration at all for the intentionality of agents. Also, norms are reduced to the description of the agents dialogical framework.

At (late) requirements time, Gaia [5] has the interesting concepts of permission and responsibility associated to each role. A permission defines the resources that (an agent playing) a role is authorized to use, and, implicitly, also the resources that it cannot use. Responsibilities are the actual functionalities of the role. Gaia, again, gives only marginal space to agents rationality, and organizational objectives are hardly distinguished from agents’ goals, which are replaced by responsibilities. As a consequence, by example, Gaia “is not intended for systems that admit the possibility of true conflicts” [14]. KAOS [8], [9] is a requirements specification methodology, which allows to describe agents goals and to refine them into the actual responsibilities. Goals are non-operational, i.e. there are no actions that can be performed to achieve them; instead, they are operationalized into the agents' behavior through constraints, which specify how goals have to be fulfilled. KAOS has the capability of capturing non-functional requirements. However, its norm awareness is limited to the concept of constraint, making hard to capture the organizational settings that generates the norms; similarly, there is no explicit support to social modeling constructs, such as the concept of strategic dependency.

The Trust-Confidence-Distrust (TCD) framework [12], [15] analyzes social networks, and takes into account the importance of trust with individuals and confidence in the network as a whole for agents' strategic dependencies. The TCD is operationalized as an extension of the *i** [24] framework. The success of the network is built on top of rules, which specify the preconditions for goals to be satisfied as well as for tasks to be performed. This approach is useful to analyze the efficiency of a trust-based network of agents, but does not allow to capture the nature of the norms and their organizational scope.

A security-enhanced version of the Tropos [4] methodology, SecureTropos [13], builds the security framework on the concepts of services ownership and delegation. In order to ensure access control, strategic dependencies are refined as the sum of a trust condition and a delegation. For instance, the delegation of permission is perceived by the delegatee as having the permission to fulfill a service, whereas the delegation of execution is conceived as giving to the delegatee charge to get the service fulfilled. This approach has the limitation that is peer-based; it means that each agent autonomously decides which service to delegate and to whom, according to (subjective) trust parameters. Therefore, the concept of obligation and permission are weak and not related to an overall organizational perspective.

A different approach to early requirements that stresses the importance of the regulatory context comes from the works on semantic analysis of laws' texts [2], [3]. Obligations, rights and permissions are extracted and represented as goals for the system-to-be. The technology being developed is a very interesting support for RE, even if the chosen representation of laws (GBRAM) seems to suffer of the same drawbacks as above.

III. Rationality and Normativeness

Agents' behavior can be described along two dimensions: goal-based behavior or rule-based behavior [17]. In the first case, we speak about Rationality; in the second case, we speak about Normativeness. According to how much agents are led by rationality or normativeness, agents organizations can also have different structures. They are defined markets, if agents are essentially strategic and utilitarian; networks, if autonomous agents act in a context of strategic interdependencies; hierarchies, if normativeness is exploited in order to articulate agents behavior so that it is compliant with hierarchically higher requirements [20]. A more fine-grained taxonomy of organizations can be seen in fig. 1.

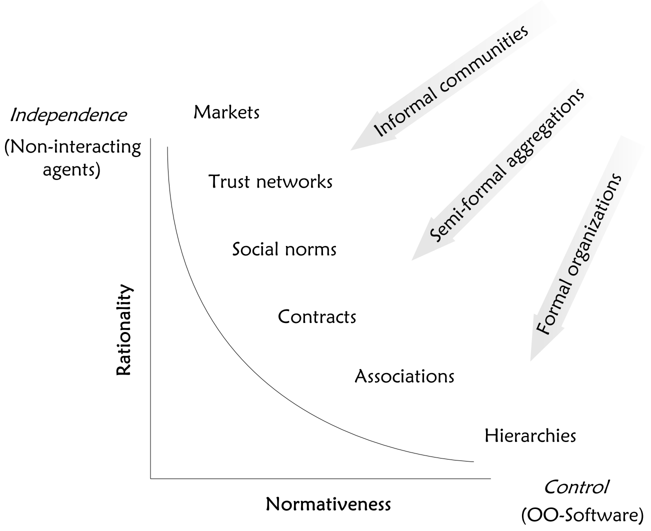

Fig. 1. Rationality-Normativeness trade-off. We classify organizations on the basis of the behavior of their stakeholders. If they are purely rational, we have a market. If they are under control of an authority, we have a hierarchy. Mixed forms are also possible: trust, social norms, contracts and associations are placeholders that recall artifacts of human organizations where agents rationality is altered by normative influence.

An idea, consistent with the concept of normativeness, is that successful organizational patterns can be reused instead of building new goal-driven requirements from scratch [6]. According to such idea, the organization types mentioned above can effectively be described along three distinct views:

- The *intentional view*, which focuses on the strategic objectives of stakeholders. Strategic objectives express the "why" of the behaviors, and can be represented using abstractions of intentional elements, like, goals, softgoals, tasks and resources [24], [4].
- The *social view*, which concerns the relationships and strategic dependencies among stakeholders [24], [4].
- The *process view*, which finally specifies the "how" of a control flow.

Going further into the normativeness metaphor, we analyze the notion of norm. A comprehensive ontology of norms is still missing in the literature, so we adopt the one taken from the ontology of law provided in [22]. In such paper, the author identifies three aspects of law: the norm frame, which contains the constitutive elements of the law; the act frame, which contains the elements of the legal action; and the concept-description frame, which regards the possible meanings of the legal statement. The norm frame is based on the concept of norm as the medium for "communicating standards of behavior". We observe that a standard of behavior should comprise the components of the three views described above. Therefore, by communicating a standard of behavior, a norm acts as the propagation vehicle for organizational pattern.

The normative layer, in an organizational context, can make significantly heavier the requirements elicitation, since it alters goal formation and reachability. However, we state that it is possible to exploit the same concept of norm, in order to improve the requirements elicitation process. The idea is to separate the rational and the normative concern of a system,

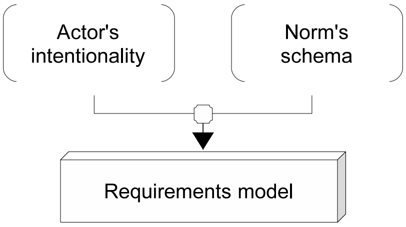

Fig. 2. Requirements model as two-layer domain.

and treat them separately. By doing so, the actual requirements domain of a normative environment can be produced as the result of merging the two concerns (see fig. 2).

IV. A CASE STUDY

In order to support our approach, we have chosen to study the Wikipedia project [23], the free online encyclopedia. It's a public and well documented project: this allows everyone to access the information about it and to verify the results. Moreover, it has some interesting characteristics: it is a network of autonomous stakeholders, but, at the same time, the project as a whole has some overall goals that are only achievable if agents spontaneously decide to pursue them. Finally, in order to obtain an overall consistency and quality of the content, editorial and behavioral policies are have to be enforced. In these conditions, the capabilities of self regulation of the organization become a critical factor.

A. The Wikipedia scenario

In one sentence, the wikipedia itself describes its structure as "a mix of anarchic, despotic, democratic, republican, meritocratic, plutocratic and technocratic elements". It seems to span over many of the organization types seen in fig. 1. We will now briefly explain such structure under the three views: intentional, social and process.

Intentional view. Wikipedia is an online encyclopedia, whose institutional goal is to provide free access and quality content. The institutional goal is refined into three main content criteria: the neutral point of view of the pages; the verifiability of the sources; the avoidance of original research content. The idea behind the project is to rely on the spontaneous and autonomous contribution of the users, which become at the same time the editors of the encyclopedia. However, because of its open nature, attackers have to be considered users as well as reliable contributors. As a consequence, a large number of regulations and policies have been created, which differentiate classes of users according to their own goals and assign to more reliable and active contributors more powers and duties.

Social view. In order to achieve its institutional goal, resources are needed, both technical (hardware and software) and economic, which are provided by donations. Resources have then to be properly administrated and spent, and this is the reason for a non-profit corporation, the Wikimedia Foundation Inc., to exists. It owns hardware and software resources, employs technical staff, and does everything else, which cannot be performed by a virtual on-line community. For these reasons, the community depends on the foundation for having resources, and the foundation depends on the community for contributing with personal capabilities (knowledge, time, ...). On the other hand, the community depends on the foundation for any legal or technical decision, but, at the same time, the foundation being administrated by a "Board of Trustees", which is elected with voting by the community itself, the foundation depends on the community for making the operational decisions.

Process view. The very basic, active tasks in the encyclopedia are the creation, modification, and deletion of content. In principle, any user who wants to edit some content to the encyclopedia can do it. However, in order to preserve the objectives seen from the intentional point of view, not everything can be allowed at any time and to anybody. Procedures, known as policies, have been created, which, once applied, should lead the goals of the project to be reached. For instance, a user should be "bold" when adding content, should follow the quality standards provided by the community or do the proper request before doing some actions. Moreover, the policies exploit the social structure of the community, and the strategic dependencies among stakeholders. If a user finds that a page does not comply with the overall quality standard, he can't directly delete the page, since he depends on the administrators to do this, so a request for deletion must be done to users playing the "administrator" role.

B. Considerations

The interesting part here is the nature of process. Individuals' actions are evaluated according to their conformance to guidelines, policies and norms. On the one hand, this evaluation influences the behavior of the stakeholders, which can decide to comply or to violate them. On the other hand, such guidelines and norms come from the community, and the community decisions arise from the behavior and opinions of the stakeholders (consensus).

Despite to its open nature, it has been demonstrated [16] that the quality of the wikipedia content is consistent with other expert-created encyclopedias. Two factors have been identified as explanations of this fact:

- Appropriation. "Wikipedia users appropriate norms and expectations about what an 'encyclopedia' should be, including norms of formality, neutrality, and consistency, from the larger culture" [11].
- Enforcement. "Those norms are enforced through the agency of dedicated, socially-approved members of the Wikipedia community" [11].

In next section we focus on the first aspect, explaining the why and how of its use.

V. METHODOLOGY

The concept of "Appropriation" can play an important role since it describes the way agents interiorize regulations. In other words, it is the core mechanism that causes agents' intentionality to be modified by the regulatory context. The concept of appropriation can be used to make more effective organizational and business modeling.

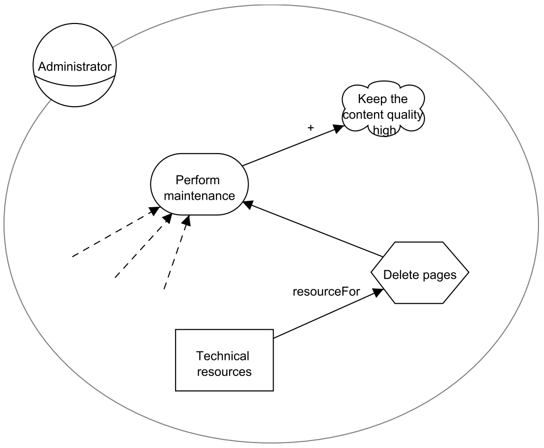

Fig. 3. The rationale of the Administrator: he wants the content quality to be kept high; therefore, he is responsible to perform maintenance; this is means-ended by (among others) the task of deleting bad pages; and special resources (i.e., administration password) are needed to perform such task.

Scenario. We have taken one particular policy from the depicted scenario, namely, the page deletion policy (PDP). It describes which pages, and how, have to be deleted. The policy applies after a user has inserted a new page or when a user makes a request for deletion. The administrator should check the quality of the page, according to some criteria (specified in other policies). If the page respect the quality standards, the page is kept. If the page has no content at all (garbage like vandalism or nonsense), the page should be immediately removed. If the page should be deleted, but it's not clearly "garbage", it is moved to the Category:Article_for_Deletion. In this case, a discussion occurs, which involves any user that has something to say. Every participant to the discussions gives a vote (keep or delete). And if, after five days of open discussion, a rough consensus occurs about the deletion, an administrator deletes the page. If the consensus occurs about keeping the page, the page is kept. A policy of the encyclopedia is that, if no clear consensus emerge, the page should be kept.

A. behavior of the Actors

In order to use the "appropriation" notion, at least a stakeholder (actor) has to be identified, which the policy applies to. The actor can be modeled using an Agent-Oriented Software Engineering (AOSE) formalism. We have chosen the *i** formalism, since it can represent effectively both the actor's rationale (the view on the strategic objectives of the actor, the actor's plans or tasks to reach the goals, and the resources he needs to activate the plans), as well as the actor's strategic dependencies (refer to [24] for a complete description of the *i** notation).

Scenario. A skeleton of *i** rationale for the actor "Administrator" is depicted in fig. 3. The administrator (represented as a Role) has the goal "perform maintenance", which partially contributes (arrow with the "+") to the softgoal "keep the content quality high". Among the others (the dotted lines mean that the actor has other tasks and goals; for simplicity, we focus only on a part of its rationale), a particular task ("Page deletion") is in "means-end" relation (simple arrow) with the goal. The task requires ("resourceFor" relation) some "technical resources" (the rectangle), e.g., administration privileges.

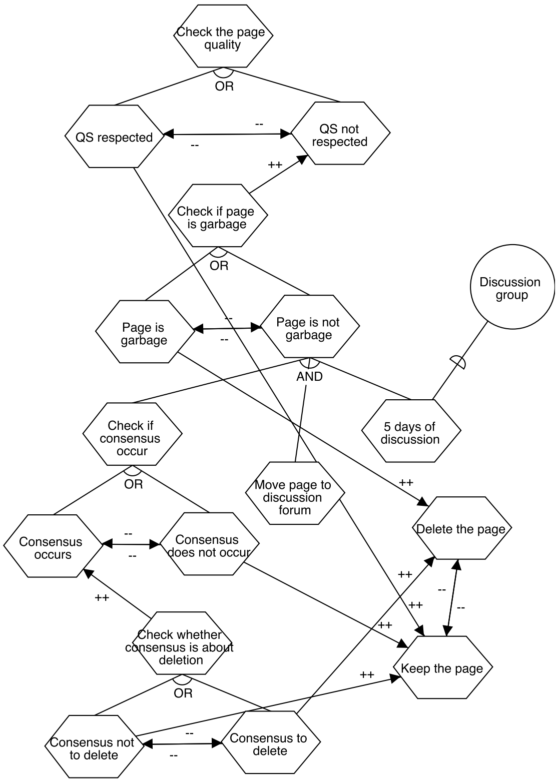

Fig. 4. The representation of the Page Deletion Policy.

B. Schema of the Norms

The PDP describes what an administrator should do in order to delete a page. Our next step in order to study the impact of the policy on the actor, is to give a formal representation to the "content" of the policy. We define the content of the policy as the schema of the norm. In order to be consistent with the actor's description given above, we adopt the same notation. The norm's schema is built with a five-steps procedure.

Step 1. Detect the basic actions that are addressed by the policy.

Scenario. In our scenario we identified 8 actions (see fig. 4), namely: [Check the page quality], [Check if page is garbage], [Move page to discussion forum], [5 days of discussion], [Check if consensus occur], [Check whether consensus is about deletion], [Keep the page] and [Delete the page]. We have represented such actions as tasks, since the goals of these actions are not specified inside this policy.

Step 2. Identify the bifurcations. Whenever the policy presents an if-then or an if-then-else construct, we have a bifurcation in the rationale. Bifurcation means different, mutually exclusive alternatives. We represent bifurcations as task-decompositions where each of the sub-tasks represent an alternative. However, in the adopted notation there is no support for XOR-decompositions. Instead, we use an OR-decomposition plus a double negative contribution.

Scenario. We can find four bifurcations: [Check the page quality], [Check whether occurs consensus for deletion], [Check consensus is about occur] and [Check if page is garbage]. For each of them, we introduce two additional tasks, representing the fact that the if-then condition is satisfied or not. For instance, for [Check the page quality] we create the tasks identified as [Quality Standard respected] and [Quality Standard not respected].

Step 3. Identify sequences. In some cases, the policy simply enumerates a sequence of steps that have to be performed. We model these sequences as task AND-decompositions, meaning that all the sub-tasks have to be performed in order to perform the super task.

Scenario. In our scenario, we have only one sequence: if a page does not respect the quality standards ([QS not respected]), but it is not clearly garbage ([Page is not garbage]), then three further actions have to be performed: [Move page to discussion forum], [5 days of discussion], and [Check if consensus occur].

Step 4. Establish contributions. When an if-then-else occurs, different further actions have to be taken. Such further actions are linked to the identified alternatives via contributions.

Scenario. The action [Delete the page] can be performed in two cases ([Page is garbage] and [Occurs consensus about deletion]), whereas the [Keep the page] has to be performed in three cases ([QS respected], [Consensus does not occur] and [Occur consensus to keep the page]). These cases are quite intuitive. Not so for the other contributions, e.g., [Check if page is garbage] contributes to [QS not respected]. The direction of the contribution is counter-intuitive. In fact, the text of policy says that "*if* the quality standard is not respected, *then* check if page is garbage". However, we are not looking at the process view but at the intentional view, which does not support the concept of flow. Thinking in terms of the rationale of the policy, it is consistent to say that the action of checking if page is garbage should imply that the quality standard is not respected.

Step 5. Establish dependencies. Finally, it is important to express the dependencies that the policies imposes. Dependencies arise when the policy cites other actors as involved in one (or more) tasks.

Scenario. Looking at the policy text, what we have represented as [5 days of discussion] is a task that is done by a generic group of people, which participate to the discussion. So the rationale of the policy explicitly consider the existence of another actor ([Discussion group], in the figure).

C. Appropriation

Once we have modeled the rationale of the actor and the rationale of the policy separately, the actual behavior of the actor can be obtained by merging the two rationales. Merging the models is the "appropriation" mechanism, and is accomplished in two steps. First, the intentionals of the norm (goals, tasks and resources, as well as their relationships) are assigned to the actor's rationale. Second, "duplicate" intentionals are identified and joint; i.e., one of them is deleted and their relationships assigned to the other one. The result of merging is a new rationale, which contains the actor's rationale, completed by the policy's prescriptions and eventually modified according to the policy itself.

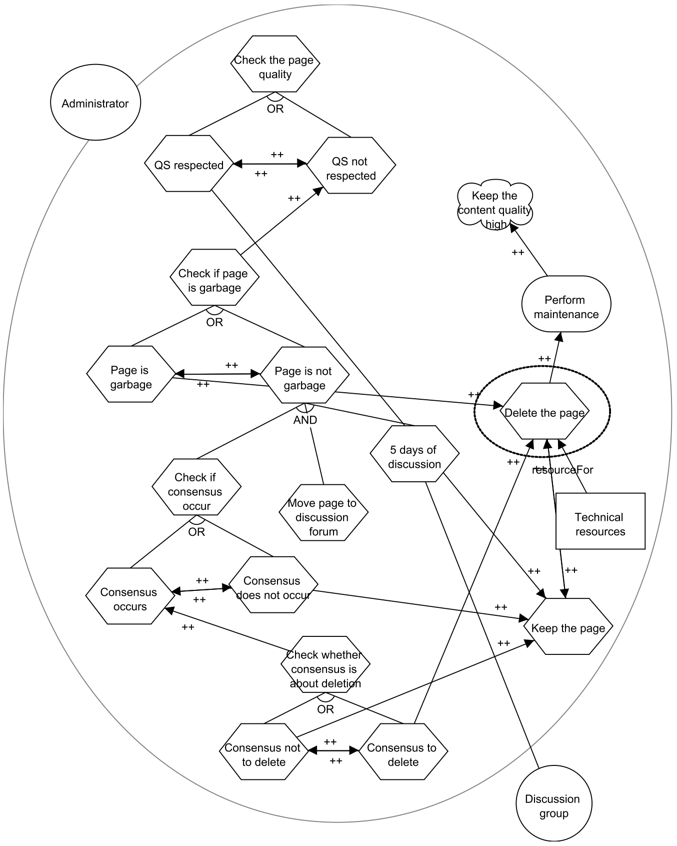

Fig. 5. The effects of Appropriation.

Scenario. The results of such operation can be seen in fig. 5. The matching among intentionals is performed on string matching basis, so that [Delete pages] in the administrator's rationale is considered the same task as [Delete the page] in the policy. By applying the policy, we know that, even if the administrator has the capability to perform the page deletion task, such task is actually activated only in circumstances specified by the policy.

D. Results and discussion

We obtain some interesting results from this work.

- Starting from the source of the policy (i.e., from its semantic), it is possible to extract a model of the norm that represents a template for the behavior of the norm's addressee.
- The rationale of the actor is altered by the schema of the norm when he appropriates it. I.e., according only to the agent's rationality, he would adopt different behaviors. The way the norm alters the actor's behavior is formalizable and, therefore, can be object of study.
- The norm also modifies the strategic dependencies among agents, which is not a-priori intrinsic in the rationales of the agents. By this way, the norm modifies the topology of an agents' organization.

Fig. 5 shows that the proposed approach allows to model the requirements of a policy-compliant actor. An important benefit of this approach is that the separation of the normative requirements from the strategic requirements reduces the complexity of the models; this involves less time spent in the design phase and a decreased probability of mistakes. Secondly, having the normative layer in a separated model means that it is possible to reuse it and apply it to many different requirements systems; this is particularly useful for dealing with recurrent problems, such as legal constraints. A third benefit is that, if the resulting requirements system looks inappropriate, it is possible to modify only the the schema of the norm and improve it.

Difficulties in this approach don't miss. We see three major problems. A critical one is the extraction of a norm's schema from its source, since a wrong schema could invalidate the reliability of the resulting requirements model. Also, the matching between intentionals (like goals and tasks) in the "appropriation" phase could also cause problems, as little differences in their names could cause wrong matchings. Finally The rules of the appropriation need to be carefully designed and tested.

VI. Conclusions and future work

In this paper we have isolated a problem of software engineering, and in particular for goal-oriented requirements engineering. The problem arise with the necessity of modeling autonomous, goal-driven stakeholders into a regulatory context that alters their natural goal formation. We deal with such problem by modeling separately the rational and the normative layer, and then by joining them with precise rules. In our view, the approach is promising in terms of capability to ease and speed the modeling phase, and (future) possibility to support with analysis some activities that are currently not supported. However, much work has to be done and is currently ongoing, in particular, we aim at:

(i) Formalize the methodology. In this paper we have presented an informal process to engineer normative environments in a more effective way. The lack of formalism makes the methodology not usable for real applications, so a rigorous formalization is needed. Moreover, formalization also generates the possibility to tool-support the methodology. *(ii)* Build an ontology of norms. The notion of norm discussed in this paper is abstract and groups many other deontic-related entities, such as policies, regulations, laws, social norms and so forth. Most likely, such different entities have different properties and present different problems. So, understanding them is an important step in order to understand if, and how, our approach can be widely applicable. *(iii)* Develop reasoning techniques. The validity of our approach can be ultimately proved when we will be able to automatically derive new information from a model. The goal is to check the overall consistency of a normative environment, detect conflicts between stakeholders' rationality and the regulatory context, verify the goals reachibility under normation, and explore alternative normations to improve to model.

References

[1] Guido Boella and Leendert W. N. van der Torre. Delta: The social delegation cycle. In *DEON*, pages 29–42, 2004.

[2] Travis D. Breaux and Annie I. Anton. Analyzing goal semantics for rights, permissions, and obligations. In *RE '05: Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE'05)*, pages 177–188, Washington, DC, USA, 2005. IEEE Computer Society.

[3] Travis D. Breaux, Matthew W. Vail, and Annie I. Anton. Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. In *RE'06: Proceedings of the 14th IEEE International Requirements Engineering Conference (RE'06)*, pages 49–58, Washington, DC, USA, September 2006. IEEE Society Press.

[4] Paolo Bresciani, Paolo Giorgini, Fausto Giunchiglia, John Mylopoulos, and Anna Perini. Tropos: An agent-oriented software development methodology. *Journal of Autonomous Agents and Multi-Agent Systems*, 8(8):203 – 236, May 2004.

[5] Luca Cernuzzi and Franco Zambonelli. Dealing with adaptive multi-agent organizations in the Gaia methodology. In *AOSE*, 2005.

[6] Enzo Colombo and John Mylopoulos. A multi-perspective framework for organizational patterns. In *25th International Conference on Conceptual Modeling (ER2006)*, 2006.

[7] Rosaria Conte, Cristiano Castelfranchi, and Frank Dignum. Autonomous norm-acceptance. In Jörg Müller, Munindar P. Singh, and Anand S. Rao, editors, *Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98)*, volume 1555, pages 99–112, 1999.

[8] Anne Dardenne, Axel van Lamsweerde, and Stephen Fickas. Goal-directed requirements acquisition. *Science of Computer Programming*, 20(1-2):3–50, 1993.

[9] Emmanuelle Delor, Robert Darimont, and Andr Rifaut. Software quality starts with the modelling of goal-oriented requirements. In *16th International Conference Software & Systems Engineering and their Applications*, 2003.

[10] Frank Dignum. Autonomous agents and social norms. ICMAS'96 Workshop on Norms, Obligations and Conventions, 1996.

[11] William G. Emigh and Susan C. Herring. Collaborative authoring on the web: A genre analysis of online encyclopedias. In *HICSS*, 2005.

[12] Gnter Gans, Matthias Jarke, Stefanie Kethers, Gerhard Lakemeyer, Lutz Ellrich, Christiane Funken, and Martin Meister. Requirements modeling for organization networks: A (dis-)trust-based approach. In *the 5th IEEE International Symposium on Requirements Engineering (RE01)*, Toronto, Canada, August 2001.

[13] Paolo Giorgini, Fabio Massacci, John Mylopoulos, and Nicola Zannone. Modeling social and individual trust in requirements engineering methodologies. In *Proceedings of the Third International Conference on Trust Management (iTrust 2005)*, volume 3477 of *Lecture Notes in Computer Science*, pages 161–176, Rocquencourt, France, 23-26 May 2005. Springer-Verlag GmbH.

[14] Nicholas R. Jennings, David Kinny, and Michael Wooldridge. The Gaia methodology for agent-oriented analysis and design. *Journal of Autonomous Agents and Multi-Agent Systems 3 (3)*, pages 285–312, 2000.

[15] Stefanie Kethers, Günter Gans, Dominik Schmitz, and David Sier. Modelling trust relationships in a healthcare network: Experiences with the tcd framework. In *ECIS*, 2005.

[16] Andrew Lih. Wikipedia as participatory journalism: Reliable sources? metrics for evaluating collaborative media as a news resource. In *5th International Symposium on Online Journalism*, April 2004.

[17] James G. March. *A primer on decision making : how decisions happen*. The Free Press, 1994.

[18] Pablo Noriega. *Agent-Mediated Auctions: The Fishmarket Metaphor*. PhD thesis, Universitat Autonoma de Barcelona, 1997.

[19] Douglass C. North. *Institutions, Institutional Change, and Economic Performance*. Cambridge University Press, Cambridge, 1990. Pp. 152.

[20] Walter W. Powell. Neither market nor hierarchy: network forms of organization, 1990.

[21] Carles Sierra, Juan Antonio Rodrguez-Aguilar, Pablo Noriega, Marc Esteva, and Josep Llus Arcos. Engineering multi-agent systems as electronic institutions. *European Journal for the Informatics Professional*, 4, 2004.

[22] Robert van Kralingen. A conceptual frame-based ontology for the law. In *First International Workshop on Legal Ontologies*, 1997.

[23] Wikipedia website. http://www.wikipedia.org/.

[24] Eric Siu-Kwong Yu. *Modelling strategic relationships for process reengineering*. PhD thesis, University of Toronto, 1995. Adviser-John Mylopoulos.