

Enhancing Goal-based Requirements Consistency: an Argumentation-based Approach

Isabelle Mirbel¹ and Serena Villata²

¹ Université de Nice Sophia Antipolis

`Isabelle.Mirbel@unice.fr`

² INRIA Sophia Antipolis

`serena.villata@inria.fr`

Abstract. Requirements engineering research has for long recognized the leading role of goals as requirement artifacts during the requirements engineering specification processes. Given the large number of artifacts created during the requirements specification and the continuous evolution of these artifacts, reasoning about them remains a challenging task. Moreover, the rising complexity of the target domain under consideration during the requirements engineering process as well as the growth of geographically distributed projects explain why the number of collected requirements as well as their complexity also increase. In this context, providing support to stakeholders in achieving a common understanding of a set of goal-based requirements, in consolidating them and keeping them consistent over time is another challenging task. In this paper, we propose an approach to detect consistent sets of goal-based requirements and maintain their consistency over time. Our approach relies on argumentation theory which allows to detect the conflicts among elements called arguments. In particular, we rely on meta-argumentation, which instantiates abstract argumentation frameworks, where requirements are represented as arguments and the standard Dung-like argumentation framework is extended with additional relations between goal-based requirements.

1 Introduction

Requirements engineering (RE) research has for long recognized the leading role of goals during the requirements engineering processes. Several goal-oriented requirements engineering approaches have been proposed in the literature [26, 10, 2, 23]. Goals have shown to be useful for achieving requirements completeness, avoiding irrelevant requirements, explaining requirements to stakeholders, structuring complex requirements documents through goal refinement, supporting decision making through alternative goal refinements, managing conflicts among multiple viewpoints, separating stable from more volatile information and driving requirements identification [19].

As it has been highlighted in Pohl [22], given the large number of artifacts created during the requirements engineering process and the continuous evolution of these artifacts, managing and organizing requirements artifacts is a

challenging task. The rising complexity of the target domain under consideration during the requirements engineering process also increases the number of collected requirements as well as their inter-dependencies and it makes this task much more challenging. Dedicated tools are required to support stakeholders in achieving a common understanding of a set of requirements, in consolidating it and in keeping it consistent over the whole project life cycle. As it has been highlighted in the literature [19], even if inconsistencies may be desirable, for instance to allow further elicitation of requirements that would have been missed otherwise, their resolution is necessary at some point. Tools are required to highlight inconsistencies and to support stakeholders who will handle the resolution process. Requirements are usually provided by different groups of stakeholders. This means that we cannot just consider a set of requirements as correct or not. When a requirement is suitable for a group of stakeholders, this leads to the removal of other requirements in order to keep the full requirements set consistent. If this requirement is discovered as not so important for another group of stakeholders, it is therefore removed from the full requirements set to insure consistency from their point of view.

Different kinds of relationships hold between goals [22]. For instance, goal decomposition relationships are distinguished from goal dependencies, i.e., two kinds of decomposition are possible depending on the fact that all subgoals are required to satisfy a super-goal (*AND-decomposition*) or at least one subgoal (*OR-decomposition*). The following dependencies have been identified [22]: *equivalence*, *conflict*, and *require*.

This paper focuses on the modeling of goal-based requirements with the aim to support the stakeholders, i.e., the agents, in detecting inconsistent sets of requirements and solving these inconsistencies. We propose to use well-known Dung-like abstract argumentation [13] to reason about the consistency of a set of goal-oriented requirements. Dung-like abstract argumentation models the information as abstract elements called *arguments*. The arguments are linked to each others by an *attack* relation. Therefore, we present a way to model additional goal-based relations, to detect the inconsistencies among goals, and provide a decision support system for their resolution.

Following the idea proposed by Bagheri and Ensan [3], we propose an approach in which consistent subsets of requirements are provided to the stakeholders to allow them to understand the different units of consistent requirements. As it has been highlighted in Bagheri and Ensan [3], the use of abstract argumentation is reasonable because it does not need the requirements to be formally defined and only needs the relationships between the requirements to be defined. With respect to the work of Bagheri and Ensan [3], which concentrates on the conflict relation, in our approach we take into consideration all the relations required to organize goals, i.e., AND/OR-decomposition, conflict, require and equivalence dependencies. Therefore, we rely on meta-argumentation [7, 8] which has been proposed as a general methodology to handle the introduction of new relations among the arguments by reusing Dung's theory and results.

Alternative approaches to argumentation theory are for instance Answer Set Programming, and first-order logic [15]. As underlined by Bagheri and Ensan [3], a drawback of pure logical formalisms for dealing with inconsistency in requirement specifications is that they identify and solve the inconsistency in pure syntactic form without taking into account the semantical information required to solve inconsistency. The advantage of using argumentation theory is twofold: first, argumentation theory provides a formal but intuitive technique to reason over inconsistency allowing the detection of the implicit relationships among the arguments and their inconsistencies, and second it allows the stakeholders to choose among different sets of consistent requirements using acceptability semantics [13] with the possibility of specifying whether the choice of the consistent set of requirements has to be guided by skeptical or credulous semantics.

The paper is organized as follows. In Section 2, we discuss why and how argumentation is helpful to check the consistency in requirements engineering. In Section 3, we detail how the different goals decomposition relationships and goals dependencies are modeled in our framework. In Section 4, we illustrate our proposal with an example. Section 5 compares the proposed approach to the related work. Finally, we conclude and give some perspectives.

2 Meta-argumentation: overview

We provide the basic concepts of Dung’s abstract argumentation [13]. A Dung-style argumentation framework [13] aims at representing conflicts among elements called *arguments*. It allows to reason about these conflicts in order to detect, starting by a set of arguments and the conflicts among them, which are the accepted arguments. The accepted arguments are those arguments which are considered as believable by an external evaluator, who has a full knowledge of the argumentation framework. A Dung-style framework is based on a binary *attack* relation among arguments, whose role is determined only by their relation to other arguments. Dung [13] presents several acceptability semantics that produce zero, one, or several sets of accepted arguments. The set of accepted arguments of an argumentation framework consists of a set of arguments that does not contain an argument attacking another argument in the set. Roughly, an argument is accepted if all the arguments attacking it are rejected and it is rejected if it has at least an argument attacking it which is accepted. The (possibly multiple) set of accepted arguments computed using one of the acceptability semantics are called *extensions*.

Definition 1 (Argumentation framework AF). *An argumentation framework is a tuple $\langle A, \rightarrow \rangle$ where A is a finite set of elements called arguments and \rightarrow is a binary relation called attack defined on A .*

A semantics of an argumentation theory consists of a *conflict free* set of arguments $cf(\mathcal{S})$, i.e., a set of arguments that does not contain an argument attacking another argument in the set. Like Baroni and Giacomin [4], we use a function \mathcal{E} called *acceptance function* mapping an argumentation framework $\langle A, \rightarrow \rangle$ to its set of extensions, i.e., to a set of sets of arguments.

Definition 2 (Acceptance function). Let \mathcal{U} be the universe of arguments. An acceptance function $\mathcal{E} : 2^{\mathcal{U}} \times 2^{\mathcal{U} \times \mathcal{U}} \rightarrow 2^{2^{\mathcal{U}}}$ is a partial function which is defined for each argumentation framework $\langle A, \rightarrow \rangle$ with finite $A \subseteq \mathcal{U}$ and $\rightarrow \subseteq A \times A$, and maps an argumentation framework $\langle A, \rightarrow \rangle$ to sets of subsets of A : $\mathcal{E}(\langle A, \rightarrow \rangle) \subseteq 2^A$.

The following definition summarizes the most widely used acceptability semantics of arguments [13].

Definition 3 (Acceptability semantics). Let $AF = \langle A, \rightarrow \rangle$ be an argumentation framework. Let $\mathcal{S} \subseteq A$. \mathcal{S} defends a if $\forall b \in A$ such that $b \rightarrow a$, $\exists c \in \mathcal{S}$ such that $c \rightarrow b$. Let $D(\mathcal{S}) = \{a \mid \mathcal{S} \text{ defends } a\}$.

- $\mathcal{S} \in \mathcal{E}_{admiss}(AF)$ iff $cf(\mathcal{S})$ and $\mathcal{S} \subseteq D(\mathcal{S})$.
- $\mathcal{S} \in \mathcal{E}_{compl}(AF)$ iff $cf(\mathcal{S})$ and $\mathcal{S} = D(\mathcal{S})$.
- $\mathcal{S} \in \mathcal{E}_{ground}(AF)$ iff \mathcal{S} is smallest in $\mathcal{E}_{compl}(AF)$.
- $\mathcal{S} \in \mathcal{E}_{pref}(AF)$ iff \mathcal{S} is maximal in $\mathcal{E}_{admiss}(AF)$.
- $\mathcal{S} \in \mathcal{E}_{stable}(AF)$ iff $cf(\mathcal{S})$ and $\forall b \in A \setminus \mathcal{S} \exists a \in \mathcal{S} : a \rightarrow b$.

2.1 Meta-argumentation

Meta-level argumentation has been proposed in several works [17, 5, 11, 21] and further developed with different goals. Boella and colleagues [7, 8], in particular, proposed the meta-argumentation methodology where extended argumentation frameworks are instantiated with meta-arguments, and reasoning in the meta-level is allowed without the need to extend Dung-like abstract framework. Meta-argumentation instantiates Dung’s theory with meta-arguments, such that *Dung’s theory is used to reason about itself* [8, 21]. Meta-argumentation is a particular way to define mappings from argumentation frameworks to extended argumentation frameworks: arguments are interpreted as meta-arguments, of which some are mapped to “argument a is accepted”, $acc(a)$, where a is an abstract argument from the extended argumentation framework EAF . Moreover, auxiliary arguments are introduced to represent, for example, attacks, so that, by being arguments themselves, they can be attacked or attack other arguments.

The function f assigns to each argument a in the EAF , a meta-argument “argument a is accepted” in the basic argumentation framework. The function f^{-1} instantiates an AF with an EAF . We use Dung’s acceptance functions \mathcal{E} to find functions \mathcal{E}' between EAF s and the acceptable arguments AA' they return. The accepted arguments of the meta-argumentation framework are a function of the EAF $AA' = \mathcal{E}'(EAF)$. The transformation function consists of two parts: the function f^{-1} , transforming an AF to an EAF , and a function g which transforms the acceptable arguments of the AF into acceptable arguments of the EAF . Summarizing $\mathcal{E}' = \{(f^{-1}(a), g(b)) \mid (a, b) \in \mathcal{E}\}$ and $AA' = \mathcal{E}'(EAF) = g(AA) = g(\mathcal{E}(AF)) = g(\mathcal{E}(f(EAF)))$.

The first step of the meta-argumentation approach is to define the set of EAF s. The second step consists of defining flattening algorithms as a function

from this set of *EAF*s to the set of all basic *AF*: $f : EAF \rightarrow AF$. The inverse of the flattening is the instantiation of the *AF*. See [8] for further details.

Definition 4. An extended argumentation framework *EAF* is a tuple $\langle A, \rightarrow \rangle$ where $A \subseteq \mathcal{U}$ is a set of arguments, and \rightarrow is a binary attack relation on A . The universe of meta-arguments is $MU = \{acc(a) \mid a \in \mathcal{U}\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in \mathcal{U}\}$, where $X_{a,b}, Y_{a,b}$ are the meta-arguments corresponding to the attack $a \rightarrow b$. The flattening function f is given by $f(EAF) = \langle MA, \mapsto \rangle$, where MA is the set of meta-arguments and \mapsto is the meta-attack relation. For a set of arguments $B \subseteq MU$, the unflattening function g is given by $g(B) = \{a \mid acc(a) \in B\}$, and for sets of subsets of arguments $AA \subseteq 2^{MU}$, it is given by $g(AA) = \{g(B) \mid B \in AA\}$.

Given an acceptance function \mathcal{E} for an *AF*, the extensions of accepted arguments of an *EAF* are given by $\mathcal{E}'(EAF) = g(\mathcal{E}(f(EAF)))$. The derived acceptance function \mathcal{E}' of the *EAF* is thus $\mathcal{E}' = \{(f^{-1}(a), g(b)) \mid (a, b) \in \mathcal{E}\}$.

Definition 5 presents the instantiation of a basic *AF* using meta-argumentation.

Definition 5. Given an *EAF* $= \langle A, \rightarrow \rangle$ where $A \subseteq \mathcal{U}$ is a set of arguments, and $\rightarrow \subseteq A \times A$. $MA \subseteq MU$ is $\{acc(a) \mid a \in \mathcal{U}\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in \mathcal{U}\}$, and $\mapsto \subseteq MA \times MA$ is a binary relation on MA such that: $acc(a) \mapsto X_{a,b}, X_{a,b} \mapsto Y_{a,b}, Y_{a,b} \mapsto acc(b)$ if and only if $a, b \in A$ and $a \rightarrow b \in \rightarrow$.

Intuitively, the $X_{a,b}$ auxiliary argument means that the attack $a \rightarrow b$ is “inactive”, and the $Y_{a,b}$ auxiliary argument means that the attack is “active”. An argument of an *EAF* is accepted iff it is accepted in the flattened *AF*.

In our approach, we propose to model decomposition relationships and dependencies as a meta-argumentation framework dedicated to goal-based requirements engineering. Goals are modeled as meta-arguments and decomposition relationships and dependencies as relations among them. Thanks to the semantics assigned to each of the decomposition relationships and dependencies introduced, mappings to the argumentation framework are possible as well as reasoning to find consistent subsets of goals, i.e., extensions of the argumentation framework. In the next sections, we illustrate our proposal with the help of an example extracted from Pohl [22].

3 Goal decomposition and dependencies

Requirements engineering is generally viewed as a process consisting of four core activities: *elicitation*, *analysis*, *negotiation* and *validation*. Each activity produces information which must be made persistent by documenting it in the right way. In order to facilitate communication, to support negotiation or to provide basis for a contract, for deriving manuals or for project planning for instance, requirements are traditionally defined in a requirement document or database. Requirements artifacts can be documented using natural language or a conceptual modeling language. Goals [19] are proposed for this purpose. Goals aim at capturing the

rationale of the software systems and document agents' intentions. Regardless of the chosen means to document requirements, they appear closely related to each other.

According to the literature review presented in Pohl [22], different kinds of relationships hold between goals. Goal decomposition relationships are distinguished from goal dependencies. Two kinds of decomposition are possible depending on the fact that all subgoals are required to satisfy a super-goal (*AND-decomposition*) or at least one sub-goal (*OR-decomposition*). With regards to dependencies, *equivalence*, *conflict*, *obstruction*, *support* and *require* relationships have been identified. *Obstruction* and *support* aim at eliciting *partial* dependencies between goals. Therefore we do not take them into account in our current framework in which we only reason on crisp acceptance, and not on partial acceptance. This is left as future work.

In our approach, we rely on the meta-argumentation methodology to formally model the requirements and these main relationships among them. We choose the meta-argumentation methodology because it allows to model extended argumentation frameworks, i.e., argumentation frameworks where additional relations among the arguments are introduced, as Dung-like abstract frameworks in order to reuse Dung's properties and theorems. We define an extended argumentation framework for reasoning about requirements as follows:

Definition 6 (Requirement-based EAF). *A requirement-based extended argumentation framework REAF is a tuple $\langle A, \rightarrow, CF, RQ, AND\text{-}dec, OR\text{-}dec, EQ \rangle$ where $A \subseteq \mathcal{U}$ is a set of requirements, $\rightarrow \subseteq A \times A$, CF is a binary conflict relation on A ($CF \subseteq A \times A$), RQ is a binary requires relation on A ($RQ \subseteq A \times A$), $AND\text{-}dec$ is a AND -decomposable relation on $2^A \times A$ ($AND\text{-}dec \subseteq 2^A \times A$), $OR\text{-}dec$ is a OR -decomposable relation on $2^A \times A$ ($OR\text{-}dec \subseteq 2^A \times A$), and EQ is a binary equivalence relation on A ($EQ \subseteq A \times A$). The universe of meta-requirements is $MU = \{acc(a) \mid a \in \mathcal{U}\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in \mathcal{U}\} \cup \{Z_{a,b} \mid a, b \in \mathcal{U}\} \cup \{R_{a,b} \mid a, b \in \mathcal{U}\} \cup \{T_{a,b} \mid a, b \in \mathcal{U}\}$, where $X_{a,b}, Y_{a,b}$ are the meta-requirements corresponding to the conflict relation ($a CF b$), $Z_{a,b}$ is the meta-requirement corresponding to the requires relation ($a RQ b$), $R_{a,b}$ is the meta-requirement corresponding to the OR -decomposable relation ($a OR\text{-}dec b$), and $T_{a,b}$ is the meta-requirement corresponding to the AND -decomposable relation ($a AND\text{-}dec b$). The flattening function f is given by $f(EAF) = \langle MA, \mapsto \rangle$, where MA is the set of meta-requirements and \mapsto is the meta-conflict relation. For a set of requirements $B \subseteq MU$, the unflattening function g is given by $g(B) = \{a \mid acc(a) \in B\}$, and for sets of subsets of requirements $AA \subseteq 2^{MU}$, it is given by $g(AA) = \{g(B) \mid B \in AA\}$.*

Roughly, the extensions of the *REAF* contain the set of requirements that do not conflict with each other and that satisfy the constraints posed by the other relations.

3.1 Goal decomposition

Two kinds of goal decomposition have been identified in the literature [22]. In the following we explain how we model them in our meta-argumentation framework.

AND-decomposition. Pohl [22] defines the AND-decomposition in the following way: The decomposition of a super-goal into a set of sub-goals is an AND-decomposition if and only if all sub-goals must be satisfied in order to satisfy the super-goal.

We model the AND-decomposition relationship in meta-argumentation as follows: a super-goal to be accepted has to have all its sub-goals accepted. The idea is that all the sub-goals are represented as meta-requirements in the meta-level. They attack meta-requirement T which attacks the meta-requirement representing the super-goal. Meta-requirement T is not connected to a real requirement in the object level, but as previously noticed for X and Y , it is just used to reason in the meta-level. The formalization of the AND-decomposition relation is presented in Definition 7.

Definition 7. *Given a REAF = $\langle A, \rightarrow, CF, RQ, AND - dec, OR - dec, EQ \rangle$, the set of meta-arguments $MA \subseteq MU$ is $\{acc(a) \mid a \in \mathcal{U}\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in \mathcal{U}\} \cup \{Z_{a,b} \mid a, b \in \mathcal{U}\} \cup \{R_{a,b} \mid a, b \in \mathcal{U}\} \cup \{T_{a,b} \mid a, b \in \mathcal{U}\}$ and $\vdash \subseteq MA \times MA$ is a binary relation on MA such that:*

- $acc(r_1) \vdash T_{r_1,a}$ iff a AND - dec r_1, \dots, r_n , and
- ...
- $acc(r_n) \vdash T_{r_n,a}$ iff a AND - dec r_1, \dots, r_n , and
- $T_{r_1,a} \vdash acc(a)$ iff a AND - dec r_1, \dots, r_n , and
- ...
- $T_{r_n,a} \vdash acc(a)$ iff a AND - dec r_1, \dots, r_n .

This is similar to the representation of a conjunctive pattern of arguments, as discussed by Villata et al. [24]. In this way, we have that the goal G is accepted only if all the sub-goals are accepted too. If one (or more) sub-goal is not accepted, then the respective meta-requirement T_i is accepted, and given the attack of this meta-requirement against the super-goal, the super-goal is made unacceptable. An example of *AND-decomposition* is shown in Figure 1, where both goals $G1$ and $G2$ need to be accepted to have goal $G3$ accepted.

Proposition 1 (Semantics of AND-dec). *Given a REAF, if it holds that a_1, \dots, a_n AND-dec b and all goals a_1, \dots, a_n are accepted then goal b is accepted too.*

Proof. We prove the contrapositive. If it holds that a_1, \dots, a_n AND - dec b and goal b is rejected, then goals a_1, \dots, a_n are rejected too. Assume a_1, \dots, a_n AND-dec b and assume that meta-requirement $acc(b)$ is rejected, then there exists at least one meta-requirement T_{a_1}, \dots, T_{a_n} that is accepted. Consequently, at least one meta-requirement $acc(a_1), \dots, acc(a_n)$ is rejected.

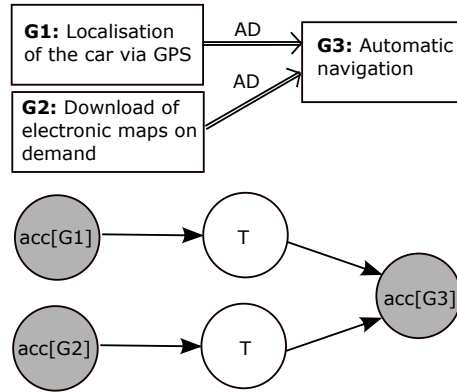


Fig. 1. Example of AND-decomposition. Accepted meta-requirements are represented in grey and rejected meta-requirements in white.

OR-decomposition. Pohl [22] defines the OR-decomposition in the following way: The decomposition of a super-goal into a set of sub-goals is an OR-decomposition if and only if satisfying one of the sub-goals is sufficient for satisfying the super-goal.

We model the OR-decomposition relationship in meta-argumentation as follows: a super-goal to be accepted needs to have at least one of its sub-goals accepted. The idea is that all the sub-goals in the meta-level attack the same meta-requirement R which attacks the meta-requirement representing the super-goal. In this way, we have that the goal G is accepted when at least one of its sub-goals is accepted too. If no sub-goal is accepted, then the meta-requirement R is accepted, and given the attack of this meta-requirement against the super-goal, the super-goal is made unacceptable. The formalization of the OR-decomposition relation is presented in Definition 8.

Definition 8. Given a REAF = $\langle A, \rightarrow, CF, RQ, AND - dec, OR - dec, EQ \rangle$, the set of meta-arguments $MA \subseteq MU$ is $\{acc(a) \mid a \in \mathcal{U}\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in \mathcal{U}\} \cup \{Z_{a,b} \mid a, b \in \mathcal{U}\} \cup \{R_{a,b} \mid a, b \in \mathcal{U}\} \cup \{T_{a,b} \mid a, b \in \mathcal{U}\}$ and $\mapsto \subseteq MA \times MA$ is a binary relation on MA such that:

- $acc(r_1) \mapsto R_{r,a}$ iff a OR - dec r_1, \dots, r_n , and
- ...
- $acc(r_n) \mapsto R_{r,a}$ iff a OR - dec r_1, \dots, r_n , and
- $R_{r,a} \mapsto acc(a)$ iff a OR - dec r_1, \dots, r_n .

An example of *OR-decomposition* is shown in Figure 2, where either goal $G1$ or goal $G2$ need to be accepted to have goal $G3$ accepted.

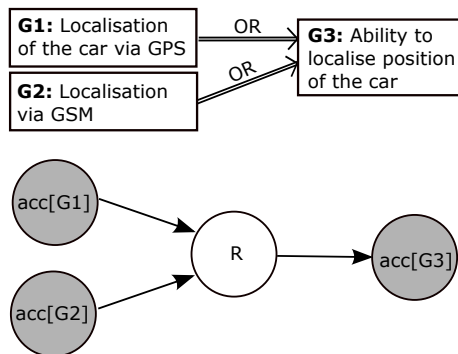


Fig. 2. Example of OR-decomposition.

3.2 Goal dependencies

Different kinds of dependencies between goals have been identified in the literature [22]. In the following, we explain how we model the *conflict*, *requires* and *equivalence* dependencies in our extended argumentation framework.

Conflict dependency. Pohl [22] defines the *conflict* dependency in the following way: a *conflict* dependency exists between two goals if the satisfaction of one goal entirely excludes the satisfaction of the other goal, and vice versa.

In Definition 9, we present how to model the conflicts among requirements.

Definition 9. Given a requirement-based extended argumentation framework $REAF = \langle A, \rightarrow, CF, RQ, AND\text{-}dec, OR\text{-}dec, EQ \rangle$, the set of meta-requirements $MA \subseteq MU$ is $\{acc(a) \mid a \in \mathcal{U}\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in \mathcal{U}\} \cup \{Z_{a,b} \mid a, b \in \mathcal{U}\} \cup \{R_{a,b} \mid a, b \in \mathcal{U}\} \cup \{T_{a,b} \mid a, b \in \mathcal{U}\}$ and $\mapsto \subseteq MA \times MA$ is a binary relation on MA such that:

- $acc(a) \mapsto X_{a,b}$ iff a *CF* b and
- $X_{a,b} \mapsto Y_{a,b}$ iff a *CF* b and
- $Y_{a,b} \mapsto acc(b)$ iff a *CF* b and
- $acc(b) \mapsto X_{b,a}$ iff a *CF* b and
- $X_{b,a} \mapsto Y_{b,a}$ iff a *CF* b and
- $Y_{b,a} \mapsto acc(a)$ iff a *CF* b .

The two meta-requirements X and Y are used to model the conflict relation in the meta-level as well as the attack relation. The semantics of the conflict dependency is similar to the semantics of the attack relation in Dung-style abstract argumentation. The difference is that the attack relation is directed from an argument to another argument while the conflict dependency leads to a cycle of attacks, i.e., the two arguments attack each other. An example of *conflict* dependency is shown in Figure 3, where goal $G1$ cannot be accepted if goal $G2$ is accepted and vice versa. In particular, the three extensions using complete semantics are $\{G1\}$, $\{G2\}$, and \emptyset .

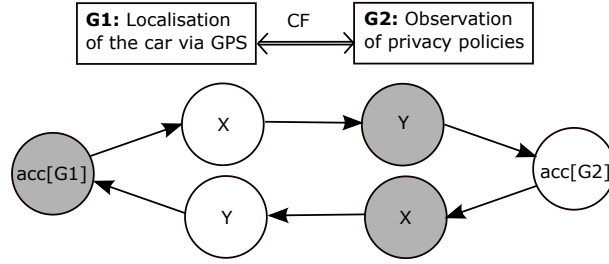


Fig. 3. Example of conflict dependency where the extension $\{G1\}$ is shown.

Require dependency. Pohl [22] defines the *require* dependency in the following way: a goal G_1 is related to a goal G_2 by a *requires* dependency if the satisfaction of the goal G_2 is a prerequisite for satisfying goal G_1 .

We model the *requires* relation as a relation such that, given that G_1 requires G_2 , G_1 is accepted only if G_2 is accepted too. This means that if G_2 is not accepted, then G_1 is not accepted either. We formalize the *requires* relation using meta-argumentation in Definition 10.

Definition 10. Given a REAF $= \langle A, \rightarrow, CF, RQ, AND - dec, OR - dec, EQ \rangle$, the set of meta-arguments $MA \subseteq MU$ is $\{acc(a) \mid a \in \mathcal{U}\} \cup \{X_{a,b}, Y_{a,b} \mid a, b \in \mathcal{U}\} \cup \{Z_{a,b} \mid a, b \in \mathcal{U}\} \cup \{R_{a,b} \mid a, b \in \mathcal{U}\} \cup \{T_{a,b} \mid a, b \in \mathcal{U}\}$ and $\vdash \subseteq MA \times MA$ is a binary relation on MA such that:

- $acc(b) \vdash Z_{a,b}$ iff $a RQ b$, and
- $Z_{a,b} \vdash acc(a)$ iff $a RQ b$.

Definition 10 highlights that goals cannot only conflict with each other but can also require the acceptability of other goals to be themselves accepted. The *requires* relation is defined following the example of the modeling in meta-argumentation of the support relation [9]. An example of *requires* dependency is shown in Figure 4, where the goal $G1$ needs goal $G2$ accepted to be accepted too.

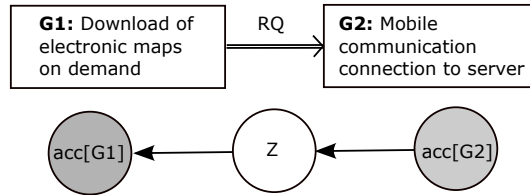


Fig. 4. Example of *requires* dependency.

Proposition 2 (Semantics of *requires*). Given a REAF, if it holds that $a RQ b$ and goal a is accepted, then goal b is accepted too.

Proof. We prove the contrapositive. If it holds that $a \text{ RQ } b$ and goal b is not accepted, then goal a is not accepted. Assume that $a \text{ RQ } b$ and assume that meta-requirement $\text{acc}(b)$ is not accepted, then meta-requirement $Z_{a,b}$ is accepted. Consequently, meta-requirement $\text{acc}(a)$ is not accepted.

Proposition 3. *Given a REAF with goals a , b and c , if there is an attack such that $a \rightarrow c$ if a RQ b and $b \rightarrow c$, then the extensions do not change, using our meta-argumentation model and one of Dung’s semantics.*

Proof. We use reasoning by cases. Case 1: $\text{acc}(a)$ is accepted, then also $\text{acc}(b)$ is accepted following Proposition 2, and given $b \rightarrow c$, $a \rightarrow c$ can be deleted without changing the extension. Case 2: $\text{acc}(a)$ is not accepted, then $a \rightarrow c$ can be deleted.

Our representation of the *requires* relation is based on the fact that a requires b is modeled by the flattening function with a path from $\text{acc}(b)$ to $\text{acc}(a)$, i.e. $\text{acc}(a)$ is accepted only if $\text{acc}(b)$ is accepted. Notice that, given $a \text{ RQ } b$, in meta-argumentation we condense all the attacks which are both on b and thus on a (both from b and thus from a) using only meta-requirement $Z_{a,b}$, as we show in Proposition 4.

Proposition 4. *Given a REAF, if there is an attack such that $c \rightarrow a$ if a RQ b and $c \rightarrow b$, then the extensions do not change, using our meta-argumentation model and one of Dung’s semantics.*

Proof. We use reasoning by cases. Case 1: $\text{acc}(c)$ is accepted, then $\text{acc}(a)$ is not accepted, follows from Proposition 2, and given $c \rightarrow b$, $c \rightarrow a$ can be deleted without changing the extension. Case 2: $\text{acc}(c)$ is not accepted, then $\text{acc}(a)$ is accepted, and the attack relation $c \rightarrow a$ can be avoided.

Goal equivalence. Pohl [22] defines the goal *equivalence* in the following way: An *equivalence* dependency exists between two goals if the satisfaction of one goal implies the satisfaction of the other goal. We model the *equivalence* dependency in the following way: given that G_1 is equivalent to G_2 then if G_1 is in conflict with other goals, then G_2 is in conflict with these other goals too, and if goal G_2 is in conflict with other goals, then goal G_1 conflicts with these goals too. In order to maintain the semantics of the *equivalence* dependency, we have to consider how to manage the conflicts addressed against goal B when goal A is equivalent to B . In this case, we want to model the situation such that every time goal B is in conflict with another goal, then this new goal is in conflict also with A and vice-versa. We achieve it by introducing an additional kind of conflict among the goals called *equivalence* attacks.

An example of *equivalence* dependency is shown in Figure 5. In this example, G_1 is conflicting with G_3 then G_3 is also conflicting with G_2 as it is shown by the dashed lines. And if a conflict involving G_2 would exist, then G_1 would also be in conflict with this goal. We do not include this case in the figure for clarity purpose.

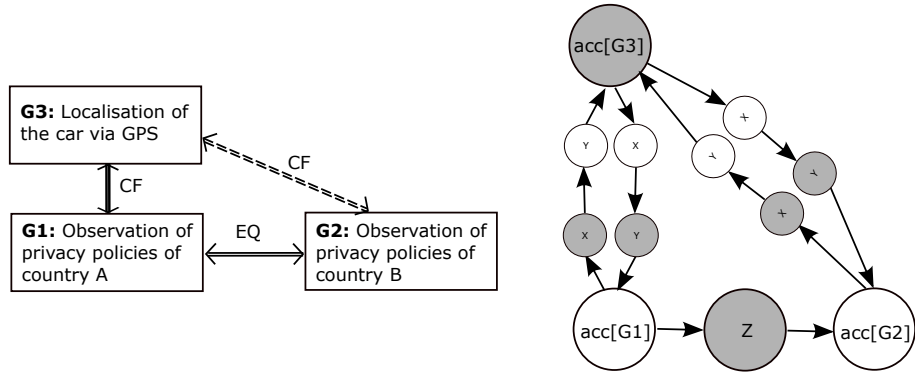


Fig. 5. Example of equivalence

In the following section, we introduce an example including the different kinds of decomposition and dependency relationships previously discussed to show the effectiveness of our approach.

4 Example

To show how the translations of the different relationships existing between goals are combined into our extended argumentation framework, an example of goal-oriented requirements modeling is presented in Figure 6.

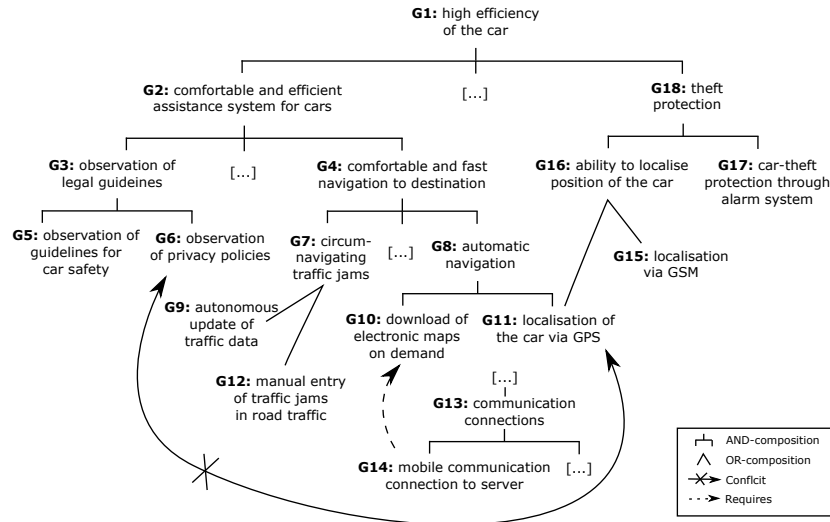


Fig. 6. Example of goal oriented requirement modeling.

In this example, there is a conflict between G_6 and G_{11} . Therefore the mapping to our meta-argumentation framework leads to 2 extensions in complete semantics: the former in which G_6 is accepted and G_{11} not accepted; and the latter in which G_{11} is accepted and G_6 is not accepted³.

Figure 7 shows the representation in the meta-level of the example from Figure 6. In this framework, we show the first extension, where G_6 is accepted and G_{11} is rejected. For clarity of the figure, we duplicated the meta-requirement $acc[G_{11}]$. The set of goals in grey corresponds to a coherent set of goals. No goal represented in white can be added to this set without making it inconsistent. Note that only meta-requirements representing the goals are requirements in the object level. Figure 8 shows the set of accepted goals on the goal hierarchy corresponding to this first extension.

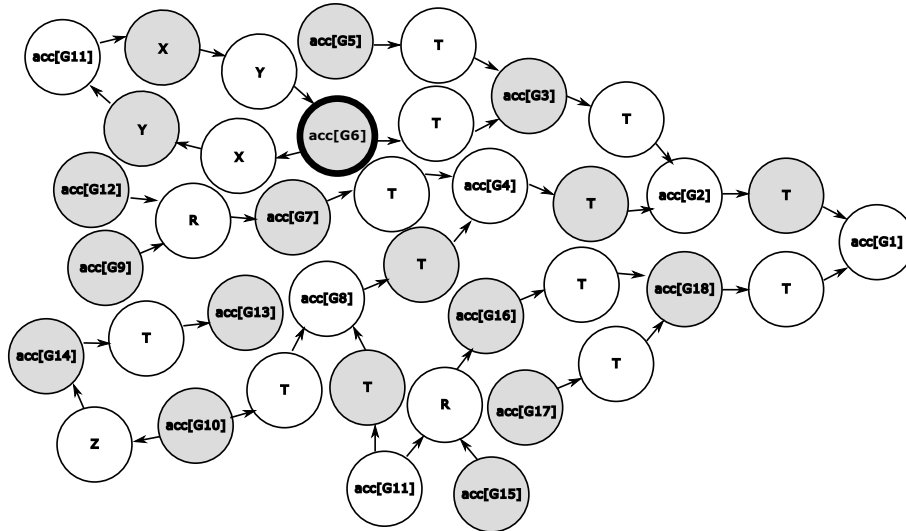


Fig. 7. The meta-argumentation framework and the extension where goal G_6 is accepted.

The second extension starting from the goal-oriented requirement modeling of Figure 6 is shown in Figure 9. In this extension, G_{11} is accepted and G_6 is not accepted. Again, meta-requirement $acc[G_{11}]$ is duplicated for clarity reasons. No goal represented in white can be added to the set without making it inconsistent. Figure 10 shows the set of accepted goals on the goal hierarchy corresponding to the second extension.

As it is shown by this running example, we propose an approach in which consistent subsets of an initial goals set are provided to the stakeholders to

³ We do not consider here the third extension \emptyset because we want to provide the stakeholders with alternatives where the two conflicting goals are included.

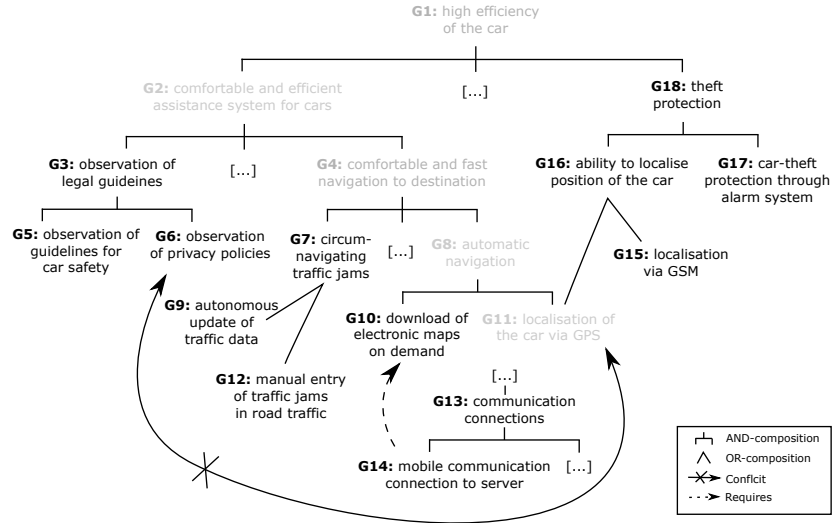


Fig. 8. The object level of the example with the goals accepted in the first extension.

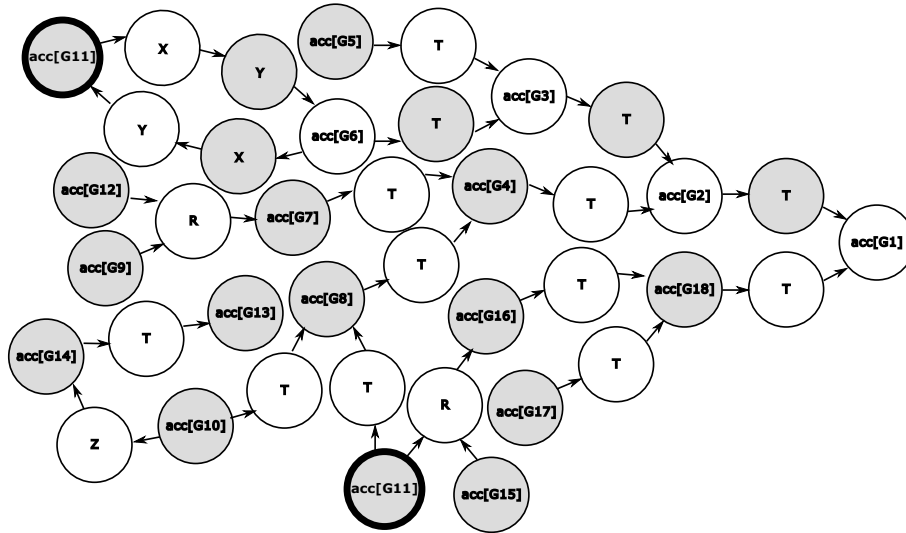


Fig. 9. The meta-argumentation framework and the extension where goal G_{11} is accepted.

allow them to understand the different units of consistent goals. This is particularly useful when the number of collected requirements as well as the number of inter-dependencies are big. In this context, our approach aims at supporting stakeholders in achieving a common understanding of a set of goal-based re-

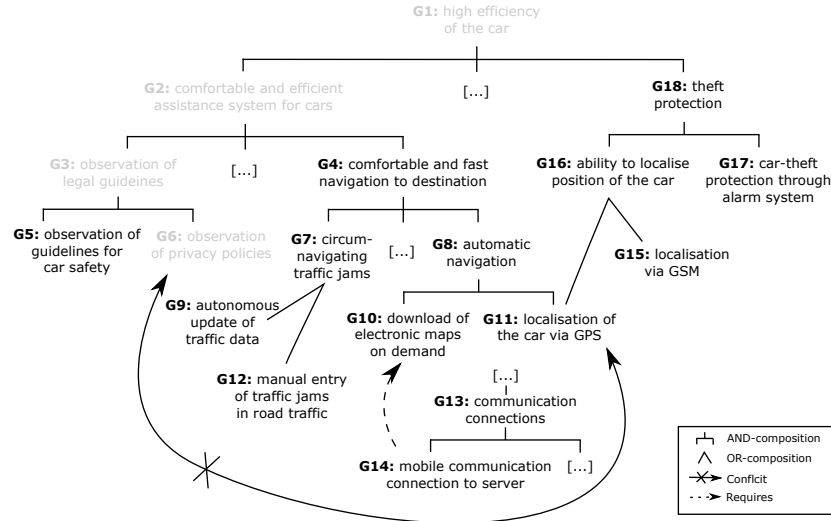


Fig. 10. The object level of the example with the goals accepted in the second extension.

quirements and at providing them a decision support system for inconsistencies resolution.

5 Related work

Frameworks to reason about goals have already proven to be useful to support goal-based requirements management. In van Lamsweerde et al. [20], for instance, goals are specified in a formal way to support reasoning on their content. In Giorgini et al. [14], an approach is proposed to analyze goal hierarchies in order to establish goals satisfiability (full, partial or none). The idea is to show the impact of the adoption of some goals on the other goals of the system. Approaches relying on argumentation have already been proposed to check the consistency of a requirement set [6, 3]. Our approach relies on meta-argumentation while the proposal of Bagheri and Ensan [3] adopts an extension of Dung's framework with preferences over the arguments. Jureta et al. [18] present a formal model to analyze the discussions between the stakeholders about the validity of the requirements engineering artifacts using argumentation theory. In particular, they introduce the Acceptability Evaluation framework (ACE) which is a propositional reasoning framework. In this framework, an acceptability condition is proposed on an artifact such that if the condition holds then it means that the relative validity for the artifact is verified. They use argumentation theory to model the discussions among stakeholders where inference, attack and preference relationships are used. There are several points which distinguish our approach from the work of Jureta et al. [18]: first, we rely on abstract argumentation theory

and on its notion of acceptability semantics to assess which are the acceptable requirements, instead of introducing a new framework; second, we reason at the pure abstract level as done by Bagheri and Ensan [3] starting from the set of goal-based requirements and their relationships, we are not interested in modeling the discussions of the stakeholders to verify the validity of an artifact using a propositional language; third, we do not consider only the conflict and the preference relationships among the requirements, but we consider goal-based requirements and their additional relationships (AND/OR-decomposition, and require, conflict and equivalence goal dependencies) provided by the stakeholders to detect the (possibly multiple) consistent sets of requirements, showing in addition how the inconsistencies may be resolved. Ingolfo et al. [16] use the ACE framework to deal with the compliance of software requirements. In Goknil et al. [15], a meta model is proposed to reason about requirements consistency. In their approach, some well-known relationships between requirements are formalized by relying on first-order logic. Thanks to this meta model, implicit relations and inconsistencies are detected. We propose to address this issue by using abstract argumentation theory in order to put in evidence consistent sets of requirements.

6 Conclusion

In this paper we present an approach to support consistency checking in goal-based requirements engineering. Our approach aims at detecting implicit relationships between the requirements and checking the possible inconsistencies among them.

Our proposal uses argumentation theory to formalize the requirement and their relationships, and to detect the inconsistencies. We represent requirements as abstract arguments and the conflicts among the requirements are the conflicts among the arguments. In particular, we rely on the meta-argumentation methodology to formally model the decomposition and dependencies which may exist between goals. We choose the meta-argumentation methodology because it allows to model argumentation frameworks with additional relations among the arguments as Dung-like abstract frameworks thus reusing Dung's properties and algorithms.

Several open challenges will be addressed as future works. First, we are developing a tool which asks the stakeholders to enter the set of requirements and the relationships among them, and it returns a graphical visualization of the requirements, as shown in Figure 6 and 7. The tool highlights the set of consistent requirements, and provides the stakeholder with the possible alternatives. These alternatives depend on the chosen acceptability semantics. Second, we plan to introduce into the framework also the relationships between agents and the requirements they propose. This has to be done to be able to reason about trust, allowing the expression of different evaluations of the acceptability of the arguments depending on the stakeholder who is proposing them [25]. Third, we plan to use fuzzy values expressing the degree of acceptability of the arguments to take into account partially satisfied goals [12], as it is necessary to model the

obstruction and support relationships. Moreover, we plan to address dynamical issues such as changes in the set of requirements and goals, which are common in most engineering projects. Finally, we will investigate the cost associated with argumentation-based approaches to software engineering.

References

1. Ab Aziz, R., Zowghi, D., and McBride, T. Towards a Classification of Requirements Relationships. 21st International Conference on Software Engineering and Knowledge Engineering, 26-32 (2009).
2. Amyot, D.; Mussbacher, G.; User Requirements Notation: The First Ten Years, The Next Ten Years. *Journal of Software* 6(5), 747-768 (2011).
3. Bagheri, E.; Ensan, F.: Consolidating multiple requirement specifications through argumentation. 26th Symposium On Applied Computing, 659-666 (2011).
4. Baroni P.; Giacomini M.: On principle-based evaluation of extension-based argumentation semantics, *Artif. Intell.*, 171(10-15), 675-700 (2007).
5. P. Baroni, F. Cerutti, M. Giacomini, G. Guida, Afra: Argumentation framework with recursive attacks, *Int. J. Approx. Reasoning* 52 (1) (2011) 19–37.
6. Besnard, P.; Hunter, A.: *Elements of argumentation*. MIT Press (2008).
7. Boella, G.; van der Torre, L.; Villata, S. On the Acceptability of Meta-arguments. *International Conference on Intelligent Agent Technology*, 259-262 (2009)
8. Boella, G.; Gabbay, D-M.; van der Torre, L.; Villata, S. Meta-Argumentation Modelling I: Methodology and Techniques. *Studia Logica*, 93(2-3), 297-355 (2009).
9. G. Boella, D. M. Gabbay, L. van der Torre, S. Villata, Support in abstract argumentation, in: 3rd International Conference Computational Models of Argument, IOS Press, 40-51 (2010).
10. Bresciani, P.; Giorgini, P.; Giunchiglia, F.; Mylopoulos, J.; Perini, A.; Modeling early requirements in Tropos: a transformation based approach. *Second International Workshop on Agent-Oriented Software Engineering*, 151-168 (2001).
11. C. Cayrol and M.-C. Lagasque-Schiex.; Coalitions of arguments: A tool for handling bipolar argumentation frameworks. *Int. J. Intell. Syst.*, 25(1):83–109, 2010.
12. C. da Costa Pereira, A. Tettamanzi, S. Villata, Changing ones mind: Erase or rewind? 22nd International Joint Conference Artificial Intelligence, 164-171 (2011).
13. Dung, P-M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2), 321-357. 1995.
14. Giorgini, P., Mylopoulos, J., Sebastiani, R.: Goal-oriented requirements analysis and reasoning in the Tropos methodology. *Agent-oriented Software Development*, 18(2), 159-171 (2005).
15. Goknil, A., Kurtev, I., van den Berg, K., and Veldhuis, J.-W. Semantics of trace relations in requirements models for consistency checking and inferencing. *Software and System Modeling*, 10(1), 31-54 (2011)
16. Ingolfo S., Siena A., and Mylopoulos J. Establishing Regulatory Compliance for Software Requirements. 30th IEEE International Requirements Engineering Conference, 47-61 (2011).
17. H. Jakobovits, D. Vermeir, Robust semantics for argumentation frameworks, *J. Log. Comput.* 9 (2) (1999) 215–261.
18. Jureta I., Mylopoulos J., and Faulkner S. Analysis of Multi-Party Agreement in Requirements Validation. 17th IEEE International Requirements Engineering Conference, 57-66 (2009).

19. van Lamsweerde, A.: Goal-Oriented Requirements Engineering: A Guided Tour. Fifth IEEE International Symposium on Requirements Engineering, 249 (2001).
20. van Lamsweerde, A.; Darimont, R.; Letier, E.; Managing conflicts in goal-driven requirements engineering. IEEE Transactions on Software Engineering, 24(11), 908-926 (1998)
21. Modgil, S.; Bench-Capon, T. J. M.; Metalevel argumentation. J. Log. Comput., 21(6):959–1003, 2011.
22. Pohl, K.: Requirements Engineering. Fundamentals, Principles, and Techniques. Springer, 2010
23. Rolland, C.; Prakash, N.; Benjamen, A.; A Multi-Model View of Process Modelling. Requirement Engineering 4(4), 169-187 (1999).
24. Villata, S.; Boella, G.; van der Torre, L.: Argumentation Patterns. 8th International Workshop on Argumentation in Multi-Agent Systems, 133-150 (2011).
25. Villata, S.; Boella, G.; Gabbay D.; van der Torre, L.: Arguing about the Trustworthiness of the Information Sources. 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 74-85 (2011).
26. Yu, E. Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. 3rd IEEE Int. Symp. on Requirements Engineering, 226-235 (1997).