RationalGRL: A Framework for Argumentation and Goal Modeling

Marc van Zee[1], Diana Marosin[2], Floris Bex[3], and Sepideh Ghanavati[4]

[1]Computer Science and Communication (CSC) University of Luxembourg
marcvanzee@gmail.com

[2]Luxembourg Institute of Science and Technology Luxembourg
diana.marosin@list.lu

[3]Information and Computing Science Utrecht University
florisbex@gmail.com

[4]Department of Computer Science Texas Tech University
sepideh.ghanavati@ttu.edu

Abstract. Goal modeling languages capture the relations between an information system and its environment using high-level goals and their relationships with lower level goals and tasks. The process of constructing a goal model usually involves discussions between a requirements engineer and a group of stakeholders. While it is possible to capture part of this discussion process in the goal model, for instance by specifying alternative solutions for a goal, not all of the arguments can be found back in the resulting model. For instance, reasons for accepting or rejecting an element or a relation between two elements are not captured. In this paper, we investigate to what extent argumentation techniques from artificial intelligence can be applied to goal modeling. We apply the argument scheme for practical reasoning (PRAS), which is used in AI to reason about goals to the Goal-oriented Requirements Language (GRL). We develop a formal metamodel for the new language, link it to the GRL metamodel, and we implement our extension into jUCMNav, the Eclipse-based open source tool for GRL. We evaluate the framework via a study among 20 subjects, showing that it is superior to traditional goal modeling with respect to completeness and coverage of goal search; size of the goal model; ease of use; explicitness in the assumptions made; and repeatability of conclusions across subjects.

Keywords Goal modeling · Argumentation · Practical Reasoning · Goal-oriented requirements engineering

1 Introduction

Requirements Engineering (RE) is an approach to assess the role of a future information system (IS) within a human or automated environment. An important goal in RE is to produce a consistent and comprehensive set of system requirements covering different aspects of the system, such as general functional requirements, operational environment constraints, and so-called non-functional requirements such as security and performance.

One of the first activities in RE are the "early-phase" requirements engineering activities, which include those that consider how the intended system should meet organizational goals, why it is needed, what alternatives may exist, what implications of the alternatives are for different stakeholders, and how the interests and concerns of stakeholders might be addressed [36]. This is generally referred to as goal modeling. Given the number of currently established RE methods using goal models in the early stage of requirements analysis (e.g., [25, 16, 15, 13, 12], overviews can be found in [30, 24]), there is a general consensus that goal models are useful in RE. Several goal modeling languages have been developed in the last two decades. The most popular ones include $i*$ [37], Keep All Objects Satisfied (KAOS) [31], the NFR framework [13], TROPOS [19], the Business Intelligence Model (BIM) [21], and the Goal-oriented Requirements Language (GRL) [22], which is part of an ITU-T standard, the User Requirements Notation (URN).

A goal model is usually the result of a discussion process between a group of stakeholders and a requirements engineer. For goal models are constructed in a short amount of time, involving stakeholders with a similar background, it is not often necessary to record all of the details of the discussion process that led to the final goal model. However, most real-world information systems - e.g., air-traffic management, industrial production processes, or government and healthcare services- are complex and are not constructed in a short amount of time, but rather over the course of several workshops. In such situations, failing to record the discussions underlying a goal model in a structured manner may harm the success of the RE phase of system development for several reasons as followed.

Floris: These four reasons are very interesting. They could be improved by mentioning references in each: 2 and 3 are now without references, which makes it less strong.

1. It is well-known that stakeholders' preferences are rarely absolute, relevant, stable, or consistent [26]. Therefore, it is possible that a stakeholder changes his or her opinion about a modeling decision in between two goal modeling sessions, which may require revisions of the goal model. If previous preferences and opinions are not stored explicitly, it is not possible to remind stakeholders of their previous opinions, thus risking unnecessary discussions and revisions. As the number of participants increases, revising the goal model based on changing preferences can take up a significant amount of time.
2. Other users who were not the original authors of the goal model may have to make sense of the goal model, for instance, to use it as an input in a later RE stage. If these user have no knowledge of the underlying rationale of the goal model, it may not only be more difficult to understand the model, but they may also end up having the same discussions as the previous group of stakeholders.
3. Alternative different ideas and opposing views that could potentially have led to different goal diagrams are lost. For instance, a group of stakeholders specifying a goal model for a user interface may decide to reduce softgoals "easy to use" and "fast" to one softgoal "easy to use". Thus, the resulting goal model will merely contain the softgoal "easy to use", but the discussion as well as the decision to reject "fast" are lost. This leads to a poor understanding of the problem and solution domain. In fact, empirical data suggest that this is an important reason of RE project failure [14].
4. It is not possible to reason about changing arguments and their effect on the goal model, and vice versa. A stakeholder may change his or her opinion, but it is not always directly clear what its effect is on the goal model. Similarly, a part of the goal model may change, but it is not possible to reason about the consistency of this new goal model with the underlying arguments. This becomes more problematic if the participants constructing the goal model change, since modeling decisions made by one group of stakeholders may conflict with the underlying arguments put forward by another group of stakeholders.

Our research goal is to apply practical reasoning and argumentation theory from Artificial Intelligence (AI) to the goal modeling process, with the expectation that doing so will help resolve issues 1-4 above. We identified several important requirements for such an argumentation theory: (1) it must have prior successful use in the field of software development, (2) it should allow to formally reason about the consistency of discussions and the goal model, (3) it must be adaptable for use in goal models, (4) the adapted theory must produce repeatable results that can be formalized and implemented, (5) a methodology must be identified that allows the adapted theory results to be used with existing goal modeling frameworks and tools, and (6) the adapted theory must identify arguments and rationales that might not have been found, or might have been lost, otherwise.

Floris: interesting! ad (1) why must the argumentation theory have prior successful use in software development? What is 'successful use'? An academic paper? A system prototype? A large-scale evaluation? Implementation in a system that is used daily in companies? ad(4) what do you mean by 'repeatable results?'

In this context, the main research question is: *Can AI theories of practical reasoning and argumentation be adapted, formalized, and combined with an existing goal modeling methodology to rationalize goal model with underlying arguments, and to reason about their consistency?* The satisfaction of the above six requirements will result in a positive answer to the research question.

1.1 Contributions

The contribution of this paper is to show that the Argument Scheme for Practical Reasoning (PRAS) fits all of the six requirements above. We briefly discuss how PRAS meets the first two requirements with an example from the medical domain, particularly in reasoning about medical treatment for a patient. **Floris:** medical domain is not software development, so how does this address req. 1? We, then, focus on issue three and four by modifying and implementing PRAS: We develop the Argument Schemes for Goal Modeling (GMAS), an extension to PRAS, which can be used to construct arguments for parts of a goal model. These arguments can be attacked with so-called "critical questions", which are natural language statements that question a certain aspect of the argument. We develop nine argument schemes and

18 critical questions. **Floris:** why does the preceding development of GMAS and the cq's address 3 in that it produces *repeatable* results? In order to reason about the consistency of the goal model with underlying arguments, we develop three *consistency conditions* **Floris:** does this not address req. 2?. We integrate GMAS with the Goal-oriented Requirements Language (GRL) and we implement it in jUCMNav, the Eclipse-based open source tool for GRL modeling. To address issue five, we develop the RationalGRL methodology that combines goal modeling with argumentation. Finally, we answer issue six positively through an empirical study **Floris:** is the explanation of the study copied from Singh et al¿? in which 20 subjects apply RationalGRL or GRL to develop a goal modeling in a information system concerning national security. The study shows RationalGRL is superior to GRL with respect to completeness and coverage of goal search; size of the goal model; ease of use; explicitness in the assumptions made; and repeatability of conclusions across subjects. As a side effect, RationalGRL requires more time than GRL, but is justified by improvements in quality.

1.2 Organization

The rest of this paper is organized as follows: In Section 2, we introduce a goal modeling language, i.e. Goal-oriented Requirements Language (GRL) [] and the practical reasoning framework PRAS. We also provide a motivation and examples for both formalisms. In Section 3, we first analyze the differences between PRAS and GRL in terms of concepts and relationships, after which we introduce the Argument Schemes for Goal Modeling (GMAS) and associated critical questions. We, then, illustrate GMAS through several examples. In Section **??**, we specify a consistency condition on the goal model and underlying arguments, while in Section 4, we provide a metamodel for GMAS and we link it to the GRL metamodel. We also discuss implementation in jUCMNav [] and the visualization details of our language.In Section 5, we explain how GMAS can be used by practitioners by introducing the RationalGRL methodology. It specifies how goal modeling and argumentation can be used to construct a GRL model. In Section 6, we describe the design of our empirical study and the findings from it. Section 7 presents related work and Section 8 discusses the conclusions and the future work.

2 Background: Goal-oriented Requirements Language and Practical Argumentation

In this section, we introduce the Goal-oriented Requirements Language (GRL) and the Argument Scheme for Practical Reasoning (PRAS).

2.1 Goal-oriented Requirements Language (GRL)

The User Requirements Notation (URN) [22], an ITU-T Standard, is one of the first modeling languages in the area of RE which aims at the standardization of visual notations to analyze functional (behavior) and non-functional requirements (NFRs). URN allows software and requirements engineers to identify requirements for a proposed or an evolving system and to review such requirements for correctness and completeness. URN combines two complementary notations: the *Goal-oriented Requirement Language* (GRL) [2] for goals and NFRs, and *Use Case Maps* (UCMs) [34] for scenarios, business processes and functional requirements. In this article, we restrict ourselves to GRL.

GRL is a visual modeling language for specifying intentions, business goals, and non-functional requirements (NFRs) of multiple stakeholders. It can be used to specify alternatives that have to be considered, decisions that have been made, and rationales for making decisions. A GRL model is a connected graph of intentional elements that optionally are part of actors. An actor ( ) represents a stakeholder of a system, or the system itself. Actors are holders of intentions; they are the active entities in the system or its environment who want goals to be achieved, tasks to be performed, resources to be available, and softgoals to be satisfied. Softgoals ( ) differentiate themselves from goals ( ) in that there is no clear, objective measure of satisfaction for a softgoal whereas a goal is quantifiable, often in a binary way. Softgoals are often more related to NFRs, whereas goals are more related to functional requirements. Tasks ( ) represent solutions to (or operationalizations of) goals and softgoals. In order to be achieved or completed, softgoals, goals, and tasks may require resources ( ) to be available.

A variety of links connect the elements in a GRL model. AND, IOR, and XOR decomposition links ( ) allow an element to be decomposed into sub-elements. Contribution links ($\longrightarrow$) indicate desired impacts of one element on another element. A contribution link has a qualitative contribution type or a quantitative contribution. Dependency links ( ) model relationships between actors.

GRL is based on $i*$ [37] and the NFR Framework [13], but it is not as restrictive as $i*$. Intentional elements and links can be more freely combined, the notion of agents is replaced with the more general notion of actors, i.e., stakeholders, and a task does not necessarily have to be an activity performed by an actor, but may also describe properties of a solution.

Motivation for using GRL

GRL has the existing tools that allow us to achieve requirements four and five of a formal argumentation theory as described in the introduction: (4) the adapted theory must produce repeatable results that can be formalized and implemented. (5) a methodology must be identified that allows the adapted theory results to be used with existing goal modeling frameworks and tools. GRL is an international standard with well defined syntax and semantics. This allows us to combine the theory of formal argumentation with GRL precisely, hereby satisfying issue 4. Moreover, GRL has an open source tool called jUCMNav, which simplifies an implementation. Since implementation is a key concern for us, the extensive tool support is a strong argument in favor of GRL. According to Amyot et al. [3], GRL has several benefits over other existing goal modeling languages, such as integration with a scenario notation (UCM), support for both qualitative and quantitative attributes (for contributions levels and satisfaction levels), there is a clear separation of GRL model elements from their graphical representation, and it provide support for providing a scalable and consistent representation of multiple views/diagrams of the same goal model (see [18, Ch.2] for an more details). Finally, URN and jUCMNav provide traceability links between concepts and instances of goal modeling and behavioral design models. This traceability enables direct impact analysis of goal changes on a design and vice versa. This traceability feature seems to be a good fit with the current research: We aim to add traceability links between intentional elements and their underlying arguments.

2.2 Argument Scheme for Practical Reasoning (PRAS)

Reasoning about which goals to pursue and actions to take is often referred to as *practical reasoning*, and has been studied extensively in philosophy (e.g. [**?**,33]) and Artificial Intelligence [**?**,4]. One approach is to capture practical reasoning in terms of arguments schemes and critical questions [33]. The idea is that an instantiation of such a scheme gives a presumptive argument in favour of, for example, taking an action. This argument can then be tested by posing critical questions about, for example, if the action is possible given the situation, and a negative answer to such a question leads to a counterargument to the original presumptive argument for the action.

A formal approach to argumentative about actions has been presented by Atkinson et al. [4], who define the Practical Reasoning Argument Scheme (PRAS), as follows.

PRAS In the circumstances R,
agent Ag should perform action A,
to achieve new circumstances S,
which will realize goal G,
which will promote some value V.

This scheme distinguishes between the state of affairs brought about by an action, the goal (i.e., the desired features in that state of affairs), and the value (i.e., the motivation why those features are desirable). Take, for instance, the example from [32]: the company (actor) does not have a standard corporate data model (current circumstances), and therefore decides to define such a model (action), leading to a new data model (new circumstances), which realizes the goal to adhere to a corporate data model, which promotes lower costs (value).

In practical reasoning, conclusions which are at one point acceptable can later be rejected because of new information. For example, if we find out that 'adhere to corporate data model' is not a realistic goal, we can argue that we should not define such a new data model. Atkinson *et al.* [4] define a set of critical questions that point to typical ways in which a practical argument can be criticized. An unfavourable[1] answer to a critical question then identifies a possible counterargument. One critical question is 'Does the goal realize the value stated?'. For our example, we can answer this question favourably with a 'yes', as one way to lower costs is to adhere to a company's corporate data model [32]. Another critical question is 'Can the desired goal be realized?'. If the answer to this is 'no' - for example, there is evidence that 'adhere to corporate data model' is not a realistic goal, then we would say the question has been answered unfavourably, and we have a counterargument: 'evidence shows that there is not much attention for adherence to a corporate data model, so it can probably not be realized'. Another way to counter an argument for an action is to suggest an alternative action that realizes the same goal or an alternative goal that promotes the same value. For example, in [32] it is argued that the action 'use data model of package applications' realizes the goal 'use

[1] Unfavourably can be either 'no', or 'yes', depending on what the question is.

canonical data model', which also promotes lower costs. This gives us a counterargument that also follows PRAS.

In argumentation, counterarguments are said to *attack* the original arguments (and sometimes vice versa). In the work of Atkinson et al. [4], arguments and their attacks are captured as an emphargumentation framework $(Args, Attacks)$, as introduced by Dung [17], where $Args$ is the set of arguments and $Attack$ is an attack relation between arguments. Figure **??** shows that an argumentation framework that includes the arguments from the above example can be rendered as a graph. The two alternative PRAS instantiations are A1 and A3. These arguments mutually attack each other, as 'adhere to corporate data model' (A1) is an alternative to 'adhere to canonical data model' (A3) and vice versa. Argument A2 attacks A1, as it questions the validity of the goal 'adhere to corporate data model'.

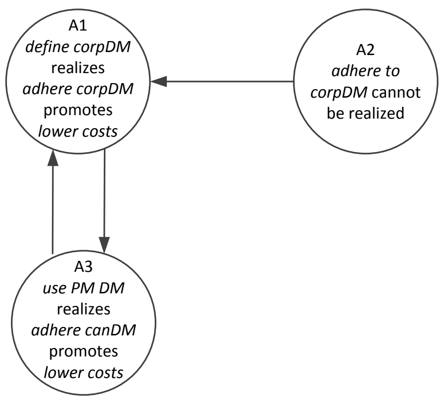

Fig. 1. Argumentation framework

Given an argumentation framework, the acceptability of arguments can be determined according to the semantics of Dung [17]. The idea is that we are justified in accepting an argument if it can be successfully defended against attacking counterarguments. In other words, an argument $A \in Args$ is *defeated* by an attacker $B \in Args$ – where $(A, B) \in Attacks$ – unless B itself can be defeated. Take, for example, the argumentation framework in Figure 1. Argument A2 is undefeated or justified, because it has no attackers. This makes A1 defeated, because one of its attackers, A2, is undefeated. A3 is then also justified, because its only attacker, A1, is defeated by A2. Thus, the set of justified (undefeated) arguments given the argumentation framework in Figure 1 is $\{$A2, A3$\}$.

attack each other, they are alternatives and without any explicit preference, it is impossible to choose between the two. However, A2 also attacks A1

where $Args$ is the set of arguments and $Attack$ is an attack relation between arguments. Dung proposes that an argument $A_1 \in Args$ is *defeated* by an attacker $A_2 \in Args$, unless A_2 itself can be defeated. Although this is an appropriate definition of reason about beliefs, it is not for reasoning about action since we are, to some extent, free to choose what we attempt to bring about. Therefore, we may choose to reject an attacker, even if it cannot be defeated, if we believe that the purpose motivating the attacked argument is more important. For instance, suppose an argument A_2 stating that a particular resource is expensive attacks an argument A_1 for using the resource. However, we may still choose to use it (i.e., accept argument A_1), if that would promote a value that we judge more importantly than expense. In order to accommodate this, *valued-based argumentation frameworks* (VAF) [10] are used. The idea is that we are given an *argumentation framework*, a set of values V, a function val mapping each argument to a value $v \in V$, and a set of *audiences* a to which the arguments are addressed. Each audience is then represented by a strict partial ordering $>_a$ on the values. Then, for a given audience $>_a$, A_2 defeats A_1 iff A_2 attacks A_1 and $val(A_2) \not>_a val(A_1)$. **Floris:** you do not give formal definitions here

It is then possible to determine which arguments in $Args$ are acceptable for an audience by determining the preferred extension. This is the maximal (under set inclusion) subset S of $Args$ such that no two arguments in S defeat each other, and all arguments in S are acceptable with respect to S, i.e., for any argument A_1 in S, if A_1 is defeated by an argument A_2 that is not in S, then there exists an argument in S that defeats A_2. The preferred extension represents the maximal consistent set of acceptable arguments with respect to the argumentation framework and a given audience or value ordering.

PRAS Medical Treatment Example

A concrete example of using PRAS comes from the medical domain, where Atkinson and colleagues use practical reasoning to formalize a system for reasoning about the treatment of a patient [7]. The system in this example consists of a group of experts in different areas, such as *treatment knowledge* and *patient knowledge*.[2] There are five critical questions relevant to the application:

CQ1 Are there alternative ways of realizing the same effects?

[2] The original application uses a single expert using different knowledge bases, but we changed this to a group of experts, each with a different expertise, in order to make the comparison with goal modeling more apparent.

CQ2 Are there alternative ways of realizing the same goal?
CQ3 Are the assumptions on which the argument is based true?
CQ4 Does performing the action have side effects which demotes some other value?
CQ5 Will the action have the effects described?

The experts can construct an argumentation framework for a medical treatment by instantiating PAS and posing critical questions CQ1-CQ5. Consider for instance a patient whose health is threatened by blood clotting. The experts start with the null option – do nothing (EA0). Then, suppose that the treatment knowledge experts suggests blood clotting can be prevented by reducing platelet adhesion, which can, assuming aspirin is not contraindicated, be achieved by prescribing aspirin. This information is used to instantiate PAS:

EA1. *Assuming no contradictions, we should prescribe aspirin, which will reduce platelet adhesion, preventing blood clotting, and so is an efficacious course of action.*

This argument is then subjected to the critical questions. CQ1 asks for alternative solutions to reduce platelet adhesion. The expert may reply by stating chlopidogrel also reduces platelet adhesion. Asking CQ2 forces the expert to seek further solutions for preventing blood clotting, which results in an alternative action, namely to administer streptokinase, which has the same goal of preventing blood clotting. These are formed into two arguments, EA2 and EA3:

EA2. *Assuming no contradictions, we should prescribe chlopidogrel, which will reduce platelet adhesion, preventing blood clotting, and so is an efficacious course of action.*

EA3. *Assuming no contradictions, we should prescribe streptokinase, which will increase blood clot dispersal, preventing blood clotting, and so is an efficacious course of action.*

These three arguments all mutually attack one another, resulting in the argumentation framework shown in figure **??**.

Each of the arguments are in mutual conflict, so it is not clear which action to pursue. This depends on the preferences of the experts. For this, suppose the experts vote and agree the effect 'reduce platelet adhesion' is a preferred means by which the goal can be realized. Moreover, they prefer aspirin to chlopidogrel because it is cheaper.

These preferences are represented in the argumentation framework by removing attacks of unfavorable actions, so EA1 is no longer attacked, and EA3 no longer attacks EA2. In this case, EA1 forms the preferred extension, and its action is currently the best candidate. However, there are still remaining critical questions that can be asked. For instance, EA1 assumes that aspirin was not contraindicated, and CQ3 instructs us to test this assumption. Suppose that the patient knowledge expert notes the patient has a history of gastritis, and infers that aspirin is contraindicated because its acidity may result in gastric ulceration. The expert forms this into an argument motivated by the value of safety:

EA4. *Where there is a history of gastritis and no acid reducing therapy, we should not prescribe aspirin, which would cause excess acidity, which would risk ulceration, and so is unsafe.*

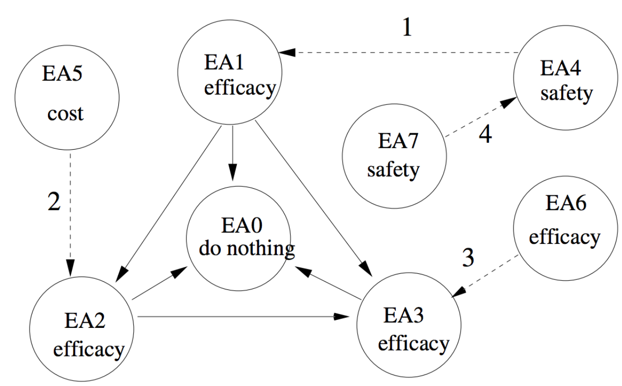

Fig. 2. Final argumentation framework showing all critiques from [7]

Floris: please explain EA5, 6 and 7 as well In a similar manner the experts construct three more arguments that attack other arguments in various ways, resulting in the argumentation framework shown in figure 2. In order to determine which arguments are accepted, the preferred extension is computed by first adding the arguments with no attackers, which are EA5, EA6, and EA7. EA7 defeats EA4, which means that EA1 can be included, since its only attacker is defeated. EA1 therefore defeats EA2 and EA4, and also excludes EA0. Thus, in the end, the winning argument is EA1, which prescribes aspirin.

Motivation for using PRAS

In the introduction, we discussed six requirements for a practical reasoning and argumentation framework to be used in our research. The first requirement is: (1) it must have prior successful use in the field of software development. The example of the previous subsection is just one of many where PRAS has been successfully used by computer scientists to formalize arguments in a computer system. **Floris:** has it been implemented? and does that mean that it has prior successful use in softw dev? The usefulness of the argument scheme approach for the analysis of practical reasoning situations has been shown in various other areas including e-democracy [6], law [9] and reasoning about the morally correct course of action [8] **Floris:** Safety critical actions, explaining character motives in stories, planning. The second requirement from the introduction is: (2) it should allow to formally reason about the consistency of arguments. The formal semantics in terms of value-based argumentation frameworks allows for the computation of accepted arguments. This semantics is extensively studied in the literature, and various different semantics exists, which lead to different flavors of acceptability of arguments. [] The third requirement from the introduction is: (3) it must be adaptable for use in goal modeling. In the next section we adapt PRAS in order to do so, but already at first glance the concepts that are used in PRAS and in goal modeling are very similar. In GRL, "tasks" can be performed by actors, while in argument scheme PAS these are called "actions". Both PAS and GRL have the concept of a "goal". Finally, the GRL notion of a "softgoal" seems semantically identical to PAS's "value", namely that it is a underlying motivation for which goals are desirable. We therefore decided that PRAS does indeed meet our first three requirements of a formal argumentation theory to be used in our research.

3 Argument Schemes for Goal Modeling (GMAS)

PRAS is not directly applicable to goal modeling in general, and GRL in particular. There are differences in the concepts used, and the relationships between these concepts. Some elements appear in PRAS that do not appear in GRL, some elements appear in GRL but not in PRAS, and some concepts appear in both formalisms but with different names. Moreover, GRL allows more relations between elements than the ones appearing in PAS. In this section, we first analyze these differences in more detail. After this, we introduce and motivate Argument Schemes for Goal Modeling (GMAS) and associated critical questions, which are an extension to PRAS. While PRAS contains a single argument scheme PAS, GMAS instead consists of nine argument schemes, which each can be used to instantiate an argument for particular part of a goal model. With each argument scheme we associate critical questions that can be used to attack arguments or provide alternatives. This enables focused discussions about elements and relationships of a goal model. In the last subsection we illustrate applications of the argument schemes and critical questions with a simple example.

3.1 Differences between GRL and PRAS

The concept that appears in PRAS and not in GRL is *circumstance*. In PRAS, an action a is performed in current circumstances R, and by performing this action new circumstances S are obtained. From this new circumstance S, another action may be performed again, and so on. In the AI literature, this type of reasoning usually referred to as "AI planning" [35]: from some initial state, find a sequence of actions such that a goal state is reached if these actions are executed. This planning approach is taken by a number of contributions using PRAS to create joint plans for a group of actors, often using a transition system as the underlying semantics [**?**]. However, in the early RE phase of an IS one generally does not consider temporal planning. Decisions about the temporal order in which tasks should be executed are generally postponed until a later stage. In other words, goal modeling is not concerned with making plans, scenarios, or business processes for the information system. In URN (see section 2.1) the distinction between goal modeling and business process modeling is clearly visible in the distinction between two separate modeling languages for the two activities. First, goal models are created in GRL, while after this use case maps (similar to business processes) are created in UCM. Therefore, when extending PRAS for GRL we do not consider the notion of *circumstances*.

The concept that appears in GRL but not in PRAS is *resource*. A resource may be used by a part of the system, or an actor, in order to perform a task in order to reach a goal. Therefore, PRAS has to be extended with the notion of a resource in order for it to be applicable to GRL.

Finally, there are two concepts appearing in both formalisms under different names. Where PRAS uses the notions "action" and "value", GRL uses the concepts "task" respectively "softgoal". However, these two concepts are semantically identical.

With respect to relationships between concepts, the argument scheme PAS from Section 2.2 contains the following relationships:

R1. Actor AG *performs* action A
R2. Action A *realizes* goal G
R3. Goal G *promotes* value S
R4. Actor AG *has* goal G
R5. Actor AG *has* value S

Relationships R4 and R5 are implicit in PAS, but follow directly from its formulation (see [4] for more details). The relationship "promotes" does not exist in GRL, but the relationship "contributes to" does, which is conceptually very similar. Therefore, we choose to use this relationship to formalize the "promotes" relation.

As mentioned in Section 2.1, GRL is a very non-restrictive modeling language, allowing intentional elements and links to be combined freely. Thus, GRL allows any links between any type of intentional element. PRAS, in contrast, only considers relationship R1-R5 above. As a result, the critical questions of PRAS are only applicable to a subset of all possible relationships in GRL. This, however, does not mean it is not possible to construct arguments and formulate critical questions for other relationship. In section 4.3, where we discuss the implementation of our tool, we explain in more detail that a user can straightforwardly add additional critical questions for other relationships as well.

3.2 The Argument Schemes and Critical Questions

Where PRAS consists of the single argument scheme PAS, our approach is to split this into a family of argument schemes, such that each separate argument scheme can be applied to a specific part of a goal model. Similar to PRAS, someone who does not accept the presumptive argument may challenge it by applying critical questions. We have modified Atkinson et al.'s original 16 questions associated with the scheme [7] by removing those questioning elements not part of GRL (*circumstances*), adding questions for additional elements GRL (*resources*), and renaming concepts and relationships (e.g., the *promotes* relationship).

We developed a total of 9 argument schemes and 18 critical questions, which are shown in table 1. Argument schemes AS1-AS4 can be instantiated for individual GRL elements, and each have a single critical question. CQ2-CQ4 are critical questions by Atkinson rephrased using the GRL terminology, while CQ1 is added in order to question a resource element, which does not appear in PRAS. AS5-AS9 are argument schemes for various GRL relationships between elements. As we mentioned before, this list is not meant to be exhaustive, since GRL does not put restrictions on any of relationships, so in theory any type of element can be combined with any type of elements, using any relationship. Here we merely use the critical questions developed by Atkinson et al.

Answering a critical question results in the creation of a new argument, which may or may not attack the original arguments, depending on which question is answered. For instance, answering CQ1 with "no" results in a new argument attacking the argument that actor a has resource R available. In contrary, answering CQ5b with "yes" does not result in an attack on the original argument, but creates a new argument stating that some other task realizes the goal as well. In general, answering a critical questions can have two effects:

1. Answering the critical question unfavorably results in a new argument attacking the original argument (*attack*).
2. Answering the critical question unfavorably results in a new argument not attacking the original argument, but representing an alternative (*no attack*).

The last column of table 1 shows for each critical question whether it results in an attack on the original argument or not.

Floris: you do not provide a formal definition of GMAS

3.3 Example: Airport Assistant

We now provide a simple scenario of arguments created by instantiating AS1-AS9, and demonstrate how new arguments can be generated through critical questions, possibly attacking the original arguments.

Consider a system for assisting users at the airport. The assistant should be able to guide users through the airport, inform them about delays, and provide other information such as the location of the check-in register for each airline company. When constructing a goal model for this system, one stakeholders argues that the user interface of the assistant should by easy to use, and that this can be realized by using a Graphical User Interface (GUI). We can formalize this as an instantiation of AS6:

A_1. *Task "Use GUI" contributes to softgoal "easy to use" (AS6).*

This argument can be subjected to critical questions. In this case, there are five critical questions that can be used on AS6, namely CQ6a-CQ6e. Critical question CQ6a asks for alternative solutions to contribute to softgoal "easy to use". One of the stakeholders may reply by stating a natural language interface is easy to use as well.

A_2. Task "Use NL interface" contributes to softgoal "easy to use" (CQ6a/AS6).

Note that A_2 does not create an attack on A_1, since both are valid tasks to contribute to the softgoal. Instead, A_2 actually generates new elements that will appear in the goal model. Now both A_1 and A_2 can be questioned. Suppose CQ6b is used by a stakeholder to argue that a side effect of using a natural language interface is that not all users may be able to understand it. When using a GUI, it may be easier to let the user change its language.

A_3. Task "Use NL interface" contributes negatively to softgoal "easy to use" because choosing a language is difficult (CQ6b).

Suppose another stakeholder uses critical question CQ6c to question A_1: she notes that implementing a GUI is probably quite expensive, because they don't have a graphical designer, so they will have to outsource this.

A_4. Task "Use GUI" contributes negatively to softgoal "low costs".

The resulting argumentation framework of these four arguments is shown in figure 3. As we can see, argument A_1 and A_4 are both not attacked by any argument. However, A_2 and A_3 are in direct conflict. Recall from sectin 2.2 that the stakeholders now have to decide about their preferences. Suppose the stakeholders decide that argument A_3 is stronger, or more preferred, than A_2. As a result, the accepted arguments are A_1, A_3, and A_4.

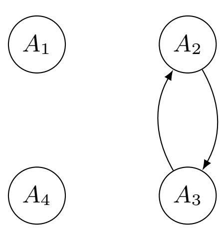

Fig. 3. Argumentation framework for Airport Assistant example.

It is now possible to translate this argumentation framework into a GRL model, which is shown in figure 4. Argument A_1 results in a contribution link from task "use GUI" to softgoal "lower costs", similarly for A_4. Since the conflict between A_2 and A_3 has been resolved in favor of A_3, there is a negative contribution link from task "use NL interface" to softgoal "Easy to use".

We can make a number of observations about the obtained GRL model and the underlying arguments:

- The underlying arguments contain information that is not represented in the GRL model. For instance, the decision to reject A_2 (i.e., that a natural language interface is easy to use) is not visible in the GRL model, but it is in the underlying arguments.
- There is a direct relation between the accepted arguments and the elements that appear in the goal model. If an element or relation has an underlying accepted argument, it appears in the goal model, while rejected ones do not appear. Therefore, if the underlying arguments change – for instance if the stakeholders change their preference from A_2 to A_3 - then the resulting goal model changes as well.
- In the current example each GRL element and relation is supported by an argument. However, this is not required. One could imagine that the stakeholders choose to add additional elements to the goal model without rationalizing them by arguments. We explain this in more detail in the RationalGRL methodology in section 5.
- Although it may be intuitively understandable how to translate argument $A_1 - A_4$ and the argumentation framework of figure 3 into the GRL model of figure 4, we have not yet formally specified how to make the translation from GMAS to GRL, and visa versa.

In this section we have fulfilled requirement three of the introduction: (3) it (the practical reasoning and argumentation theory) must be adaptable for use in goal modeling. However, we have not yet considered the fourth requirement: (4) the adapted theory must produce repeatable results that can be formalized and implemented. This is what we do in the next section by providing formal mappings from GMAS to GRL, and discussing implementation details.

4 Mapping GMAS to GRL: Metamodel and Implementation

Floris: we have no formal definition of GMAS, so proposing a formal mapping will be difficult In the previous section we adapted the practical reasoning framework PRAS for use in goal modeling, obtaining nine Argument Schemes for Goal Modeling (GMAS). These argument schemes can be instantiated to create arguments for elements and relationships in a GRL model. We introduced critical questions that can be used to construct counter arguments or alternatives, and we provided an

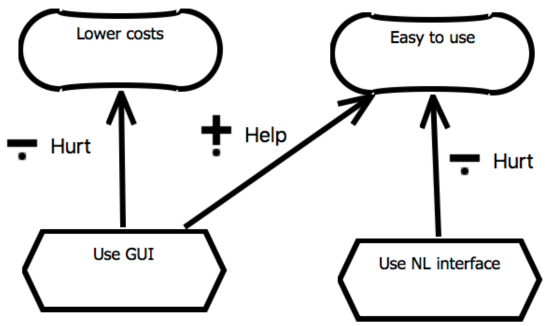

Fig. 4. GRL model for Airport Assistant example.

example in which we obtained a GRL model from a GMAS specification. However, we did not yet create a formal language for GMAS, nor did we map it to GRL. This is what we do in the current section. We provide a formal specification of GMAS in terms of a metamodel, and we link it with the GRL metamodel. Using this mapping, we detail how to translate an argumentation framework into a GRL model and visa versa. The metamodel is the starting point for the tool we developed as an extension of jUCMNav. We present this tool in the last subsection, together with the visual language we developed for our extension.

4.1 GMAS Metamodel Specification

The UML metamodel of GMAS is shown on the left hand side of figure 5.

4.2 Mappings between GMAS and GRL

Information Mappings

Formal Mappings

4.3 Implementation into jUCMNAv

5 The RationalGRL Methodology

The methodology we propose in this paper is visualized in figure 6. There are two main activities, depicted in grey, namely "goal modeling" and "argumentation". These are two separate activities that are being done in parallel.

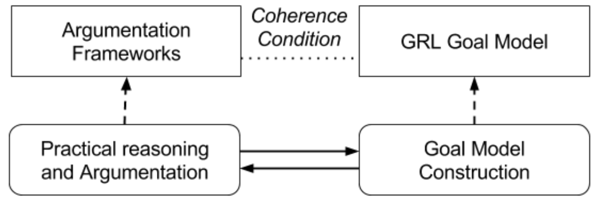

Fig. 6. The RationalGRL Methodology

6 Empirical Evaluation

7 Related Work

There are several contributions that relate argumentation-based techniques with goal modeling. The contribution most closely related to ours is the work by Jureta *et al.* [23]. This work proposes "Goal Argumentation Method (GAM)" to guide argumentation and justification of modeling choices during the construction of goal models. One of the elements of GAM is the translation of formal argument models to goal models (similar to ours). In this sense, our RationalGRL framework can be seen as an instantiation and implementation of part of the GAM. One of the main contribution of RationalGRL is that it also takes the acceptability of arguments as determined by the argumentation semantics [17] into account when translating from arguments to goal models. RationalGRL also provides tool support for argumentation, i.e. Argument Web toolset, to which OVA belongs [11], and for goal modeling, i.e. jUCMNav [28]. Finally, RationalGRL is based on the practical reasoning approach of [5], which itself is also a specialization of Dung's [17] abstract approach to argumentation. Thus, the specific critical questions and counterarguments based on these critical question proposed by [5] can easily be incorporated into RationalGRL.

RationalGRL framework is also closely related to frameworks that aim to provide a design rationale (DR) [29], an explicit documentation of the reasons behind decisions made when designing a system or artefact. DR looks at issues, options and arguments for and against the various options in the design of, for example, a software system, and provides direct tool support for building and analyzing DR graphs. One of the main improvements of RationalGRL over DR approaches is that RationalGRL incorporates the formal semantics for both argument acceptability and goal satisfiability, which allow for a partly automated evaluation of goals and the rationales for these goals.

Arguments and requirements engineering approaches have been combined by, among others, Haley *et al.* [20], who use structured arguments to capture and validate the rationales for security requirements. However, they do not use goal models, and thus, there is no explicit trace from arguments to goals and tasks. Furthermore, like [23], the argumentative part of their work does not include formal semantics for determining the acceptability of arguments, and the proposed frameworks are not actually implemented. Murukannaiah *et al.* [27] propose Arg-ACH, an approach to capture inconsistencies between stakeholders' beliefs and goals, and resolve goal conflicts using argumentation techniques.

8 Conclusions and Future Work

In this paper, we developed and implemented a framework to trace back elements of GRL models to arguments and evidence that derived from the discussions between stakeholders. We created a mapping algorithm from a formal argumentation theory to a goal model, which allows us to compute the evaluation values of the GRL IEs based on the formal semantics of the argumentation theory.

There are many directions of future work. There are a large number of different semantics for formal argumentation, that lead to different arguments being acceptable or not. It would be very interesting to explore the effect of these semantics on goal models. Jureta *et al.* develop a methodology for clarification to address issues such as ambiguity, overgenerality, synonymy, and vagueness in arguments. Atkinson *et al.* [4] define a formal set of critical questions that point to typical ways in which a practical argument can be criticized. We believe that critical questions are the right way to implement Jureta's methodology, and our framework would benefit from it. In addition, currently, we have not considered the *Update* step of our framework (Figure **??**). That is, the translation from goal models to argument diagrams is still missing. The *Update* step helps analysts change parts of the goal model and analyze its impact on the underlying argument diagram. Finally, the implementation is currently a browser-based mapping from an existing argument diagramming tool to an existing goal modeling tool. By adding an argumentation component to jUCMNav, the development of goal models can be improved significantly.

Acknowledgments

Marc van Zee and Diana Marosin are funded by the National Research Fund (FNR), Luxembourg, by the RationalArchitecture project.

References

1.
2. D. Amyot, S. Ghanavati, J. Horkoff, G. Mussbacher, L. Peyton, and E. S. K. Yu. Evaluating goal models within the goal-oriented requirement language. *International Journal of Intelligent Systems*, 25:841–877, August 2010.
3. D. Amyot, J. Horkoff, D. Gross, and G. Mussbacher. A lightweight grl profile for i* modeling. In *International Conference on Conceptual Modeling*, pages 254–264. Springer, 2009.
4. K. Atkinson and T. Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence*, 171(10):855–874, 2007.
5. K. Atkinson and T. Bench-Capon. Taking the long view: Looking ahead in practical reasoning. In *Computational Models of Argument: Proceedings of COMMA*, pages 109 – 120, 2014.
6. K. Atkinson, T. Bench-Capon, and P. McBurney. Parmenides: Facilitating deliberation in democracies. *Artificial Intelligence and Law*, 14(4):261–275, dec 2006.
7. K. Atkinson, T. Bench-Capon, and S. Modgil. Argumentation for decision support. In *International Conference on Database and Expert Systems Applications*, pages 822–831. Springer, 2006.
8. A. C. T. Bench-Capon and P. McBurney. Automating argumentation for deliberation in cases of conflict of interest. In *Computational Models of Argument: Proceedings of COMMA 2006*, volume 144, page 279. IOS Press, 2006.
9. T. Bench-Capon, K. Atkinson, and A. Chorley. Persuasion and value in legal argument. *Journal of Logic and Computation*, 15(6):1075–1097, 2005.
10. T. J. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.
11. F. Bex, J. Lawrence, M. Snaith, and C. Reed. Implementing the argument web. *Communications of the ACM*, 56(10):66–73, 2013.
12. J. Castro, M. Kolp, and J. Mylopoulos. Towards requirements-driven information systems engineering: the tropos project. *Information systems*, 27(6):365–389, 2002.
13. L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos. *Non-functional requirements in software engineering*, volume 5. Springer Science & Business Media, 2012.
14. B. Curtis, H. Krasner, and N. Iscoe. A field study of the software design process for large systems. *Communications of the ACM*, 31(11):1268–1287, 1988.
15. A. Dardenne, A. Van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Science of computer programming*, 20(1):3–50, 1993.
16. P. Donzelli. A goal-driven and agent-based requirements engineering framework. *Requirements Engineering*, 9(1):16–39, 2004.
17. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

18. S. Ghanavati. *Legal-urn framework for legal compliance of business processes*. PhD thesis, University of Ottawa, 2013.
19. P. Giorgini, J. Mylopoulos, and R. Sebastiani. Goal-oriented requirements analysis and reasoning in the tropos methodology. *Engineering Applications of Artificial Intelligence*, 18(2):159–171, 2005.
20. C. B. Haley, J. D. Moffett, R. Laney, and B. Nuseibeh. Arguing security: Validating security requirements using structured argumentation. In *in Proc. of the Third Symposium on RE for Information Security (SREIS'05)*, 2005.
21. J. Horkoff, D. Barone, L. Jiang, E. Yu, D. Amyot, A. Borgida, and J. Mylopoulos. Strategic business modeling: representation and reasoning. *Software & Systems Modeling*, 13(3):1015–1041, 2014.
22. ITU-T. Recommendation Z.151 (11/08): User Requirements Notation (URN) – Language Definition. `http://www.itu.int/rec/T-REC-Z.151/en`, 2008.
23. I. Jureta, S. Faulkner, and P. Schobbens. Clear justification of modeling decisions for goal-oriented requirements engineering. *Requirements Engineering*, 13(2):87–115, May 2008.
24. E. Kavakli and P. Loucopoulos. Goal modeling in requirements engineering: Analysis and critique of current methods. In J. Krogstie, T. A. Halpin, and K. Siau, editors, *Information Modeling Methods and Methodologies*, pages 102–124. Idea Group, 2005.
25. L. Liu and E. Yu. Designing information systems in social context: a goal and scenario modelling approach. *Information systems*, 29(2):187–203, 2004.
26. J. G. March. Bounded rationality, ambiguity, and the engineering of choice. *The Bell Journal of Economics*, pages 587–608, 1978.
27. P. K. Murukannaiah, A. K. Kalia, P. R. Telangy, and M. P. Singh. Resolving goal conflicts via argumentation-based analysis of competing hypotheses. In *23rd Int. Requirements Engineering Conf.*, pages 156–165. IEEE, 2015.
28. G. Mussbacher and D. Amyot. Goal and scenario modeling, analysis, and transformation with jUCMNav. In *ICSE Companion*, pages 431–432, 2009.
29. S. J. B. Shum, A. M. Selvin, M. Sierhuis, J. Conklin, C. B. Haley, and B. Nuseibeh. Hypermedia support for argumentation-based rationale. In *Rationale management in software engineering*, pages 111–132. Springer, 2006.
30. A. Van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *Proc. 5th IEEE Int. Symposium on RE*, pages 249–262, 2001.
31. A. van Lamsweerde. Requirements engineering: from craft to discipline. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*, pages 238–249. ACM, 2008.
32. M. van Zee, D. Marosin, S. Ghanavati, and F. Bex. Rationalgrl: A framework for rationalizing goal models using argument diagrams. In *Proceedings of the 35th International Conference on Conceptual Modeling (ER'2016), Short paper*, November 2016.
33. D. N. Walton. *Practical reasoning: goal-driven, knowledge-based, action-guiding argumentation*, volume 2. Rowman & Littlefield, 1990.
34. M. Weiss and D. Amyot. Designing and evolving business models with URN. In *Proc. of Montreal Conference on eTechnologies (MCETECH05)*, pages 149–162, 2005.
35. D. S. Weld. Recent advances in ai planning. *AI magazine*, 20(2):93, 1999.
36. E. S. Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *Proc. of the 3rd IEEE Int. Symposium on RE*, pages 226–235, 1997.
37. E. S. K. Yu. Towards modeling and reasoning support for early-phase requirements engineering. In *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering*, RE '97, pages 226–, Washington, DC, USA, 1997. IEEE Computer Society.

Argument scheme		Critical Questions	
AS1	Actor a has resource R	CQ1	Is the resource available?
AS2	Actor a can perform task T	CQ2	Is the task possible?
AS3	Actor a has goal G	CQ3	Can the desired goal be realized?
AS4	Actor a has softgoal S	CQ4	Is the softgoal a legitimate softgoal?
AS5	Task T realizes goal G	CQ5a	Will the task realize the desired goal?
		CQ5b	Are there alternative ways of realizing the same goal? **Floris:** How do you handle OR and XOR decom
AS6	Task T contributes to softgoal S	CQ6a	Are there alternative ways of contributing to the same softgoal?
		CQ6b	Does the task have a side effect which contribute negatively to the softgoal?
		CQ6c	Does the task have a side effect which contribute negatively to some other softgoal?
		CQ6d	Does the task contribute to some other softgoal?
		CQ6e	Does the task preclude some other task which would contribute to some other softgoal?
AS7	Goal G contributes to softgoal S	CQ7a	Does the goal contribute to the softgoal?
		CQ7b	Does the goal contribute to some other softgoal?
AS8	Resource R contributes to task T	CQ8a	Is the resource required in order to perform the task?
		CQ8b	Can other resources be used to perform the same task?
		CQ8c	Is the resource required in order to perform the task?
		CQ8d	Does using the resource make other resources unavailable?
AS9	Actor a depends on actor b	CQ9	Does the actor depend on any actors?

Table 1. Argument schemes and critical questions for GRL elements (CQ1-CQ4) and relationships (CQ5-CQ16), as well as whether answering the critical question unfavorably results in an attack on the original argument (right column).

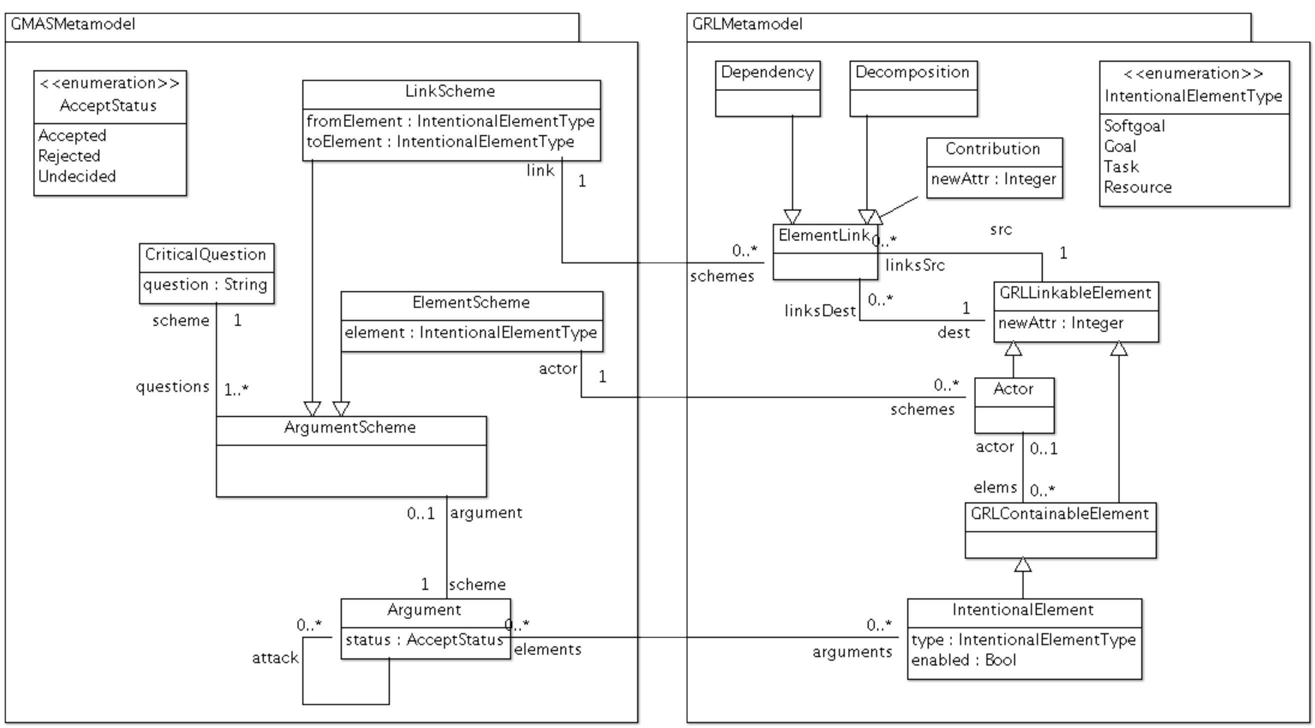

Fig. 5. Metamodel specification of GMAS and GRL