

# A Knowledge Engineering Process for the Development of Argumentation Schemes for Risk Management in Software Projects

Denise da Luz Siqueira<sup>1</sup>, Lisandra M. Fontoura<sup>1</sup>, Rafael H. Bordini<sup>2</sup>, Luis A. L. Silva<sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Ciência da Computação, UFSM. Santa Maria, RS, Brazil  
{denise.siqueira, lisandramf, silva.luisalvaro}@gmail.com

<sup>2</sup>Programa de Pós-Graduação em Ciência da Computação, PUCRS. Porto Alegre, RS, Brazil  
rafael.bordini@pucrs.br

**Abstract** – The engagement of project stakeholders in collaborative debates of risk management has an important contribution to software projects. To promote the identification, (re)use and critical analysis of stakeholders’ arguments in these debates, this paper lays out a knowledge engineering process for the development of “argumentation schemes” for risk management. This process covers activities of identification, interpretation and causal-and-effect analysis of typical risk statements. From such risk management information and reusing generalized argumentation templates from argumentation catalogues discussed in the field of Artificial Intelligence, the process leads to the specification, generalization, validation and indexing of the developed schemes. As implemented in our project, a web-based system to support the execution of these development activities allows the recording of these schemes in a semi-structured representation format. An argumentation scheme for risks of non-stable requirements is presented so as to show the reusable argumentation artifacts that can be produced when our development process is followed.

**Keywords-component:** Risk Management; Argumentation Schemes; Argumentation.

## I. INTRODUCTION

Risk management (RM) tasks act on the anticipation and treatment of risks that have a critical impact to the goals of a software project. As discussed in the CMMI standard [1], a more effective management of risks occurs when project stakeholders are engaged in the discussion of issues regarding the identification, analysis and response planning of risks. To support the (re)use of argumentation knowledge in the development of collaborative debates of RM, argumentation techniques have been exploited in the research project in which this work is situated [2-3] and in the solution of problems in other applications of Artificial Intelligence (AI) [4].

The collaborative debate of risks can be structured and mediated by argumentation-based communication protocols [5]. The exploitation of such protocols in a web-based Risk Discussion System [2-3] allows the capture and reuse of project stakeholders’ arguments. The overall idea is to maintain these arguments in a knowledge repository formed of RM experiences. In this context, a common problem is that regular project participants (e.g. clients, managers and software engi-

neers) tend to present risk analysis arguments in which relevant pieces of risk information may not be structured or even stated explicitly. In such debate situations, information to explain and justify the risks of a software project may not be presented and examined critically via users’ arguments. To approach these issues, the argumentation literature presents generalized “argumentation scheme” specifications [6] so as to capture stereotypical patterns of presumptive reasoning. In contrast with these highly generalized argumentation templates, domain-specific scheme specifications are also presented as a form of better supporting debates in selected application problems (e.g. [7-8]). Despite the significance of scheme catalogues for the development of argumentation-based tasks of knowledge engineering, there are still many forms of argument to be identified as scheme specifications. On the assumption that losing any relevant argumentation knowledge in an application problem degrades the later usefulness on what is captured, the present work lays out the main knowledge engineering activities for the specification of semi-structured varieties of argumentation schemes for requirement risks in software projects. As a result of the proposed development process, novel schemes were not only presented in [9], but they were also assessed in collaborative debates in a real-life software project aiming the design and prototyping of a virtual tactical simulator (the SIS-ASTROS project).

The remainder of this paper is structured as follows. Section II reviews background information about RM and argumentation. Section III presents our scheme development process. Section IV discusses related work and Section V reviews our contribution and identifies future work.

## II. RISK MANAGEMENT AND ARGUMENTATION

RM involves the discovery of events that can be a threat to the success of the software project so that a software development organization can plan risk response actions and build contingency reserves [10]. When analyzing failures in software projects, risks can be closely related to the poor management of requirements [10-12]. In this context, the engineering of requirements [13] is concerned with the assessment of clients’ expectations, the evaluation and negotiation of a software solution, the validations of software specifications and

the management of client needs as they are transformed into software specifications. As requirement engineering is an initial software development task, requirement risks that ended being materialized in a project are likely to have a significant impact on the project (product) goals. As reviewed in [14], the majority of such requirement specifications are described in natural language or a semi-structured version of it, where just a small portion relies on formal specification languages.

The analysis of informal arguments relies on the identification of argument components. Approaches for argumentation diagramming [15-16] allow users to organize the components of an argument not only in terms of premises and conclusion, but also via detailed elements of the argumentation model due to Toulmin [17]. The overall idea is to promote users' critical thinking and improvement of users' argumentation skills by making argumentation elements explicit in semi-formal diagrammatic models, as implemented in the Araucaria system [15], for instance. In AI, argumentation frameworks [4] are usually investigated via some appropriate version of logic formalisms of reasoning with arguments. In these settings, argumentation schemes are developed as part of a standard knowledge acquisition and representation process (e.g. [18]). Although not represented as an explicit process model, the development of such schemes involves the modelling of these semi-formal or formal argument specifications (or both).

Argumentation schemes are intended to capture presumptive patterns of reasoning [6]. The underlying idea is that such schemes could support the expression of patterns of non-deductive reasoning, or even fallacies. In our research project, users' arguments presented in the debate of RM issues are captured and reused by taking advantage of the structure of these schemes. In doing so, one exploits an argumentation scheme as a template for knowledge acquisition in the collaborative RM problem. To help the understanding of the nature of a scheme, a long-established argument scheme specifying cause and effect arguments can be presented: *Major premise*: Generally, if A occurs, then B will (might) occur. *Minor premise*: In this case, A occurs (might occur). *Conclusion*: Therefore, in this case, B will (might) occur. The evaluation (or the validity) of scheme-based arguments presented in collaborative debates is an important argumentation characteristic. In the representation of argument schemes, this evaluation involves the identification and statement of questions regarded as critical. As presented in [6], "critical questions" (QCs) for the argumentation scheme from cause to effect are: How strong is the causal relation between (X) and (Y)? (If this causal generalization is true at all) Is the mentioned evidence (X) (if there is any) strong enough to warrant the cause-effect generalization as stated? Are there other factors (F) that would or will interfere with the production of the effect (E) in this case? Is (X) the main (or single) cause for the occurrence of (Y)?

Argumentation scheme specifications for users or systems to exploit and share scheme repositories are still been investigated. To approach this issue, argumentation mark-up languages have been proposed in the context of identification and visualization of argumentation schemes [19-20]. Those pro-

posals are based on semantic web standards, which are often formalized on the basis of ontology representations. As described in [19], the Web Ontology Language (OWL) standard allows the construction of an ontology for the specification and annotation of arguments in different degrees of formalization, leading to a representation that can be processed by machines.

### III. THE DEVELOPMENT OF ARGUMENTATION SCHEMES FOR COLLABORATIVE RISK MANAGEMENT

To construct argumentation schemes not only to support the critical proposition and analysis of requirement risks, but also to promote the identification and specification of these kinds of risk-management arguments in a reusable argumentation template, a knowledge engineering process for scheme development should be put forward. Fig. 1 presents the UML activity diagram laying out our proposal for this process. To illustrate these activities, a new domain-oriented argumentation scheme developed in our project can be used: the "Argumentation scheme for risks of non-stable requirements". Such instability of requirements can be related to a volatile application domain, where clients either present new requirements in the course of a project or change requirement specifications made in the past (or both). In general, external factors of a software project can lead to new requirements or changes in the requirement specifications. To execute the development activities, a web-based system was implemented in our project (Fig. 2 - A). The activities of that process are as follows.

*To select risk statements*: it aims to compile a list of related requirement risk statements (here taken as arguments), to be structured as a reusable task-oriented argumentation scheme specification. To do so, the input artefacts are information repositories containing typical risk statements (or risk factors) [12, 21] in software projects. In addition to risk factors, concrete experiences of argumentation-based RM obtained in the past can be exploited in this activity. Moreover, development steps can be directed to the consultation with field experts in RM aiming to identify what risk factors are most subject to pro and con stakeholders' arguments. An example of such typical risk statements refers to arguments related to the proposition of risks about non-stable requirements. These kinds of requirement risk statements are the output artefact of this activity.

*To describe risk interpretations*: it aims to exploit risk interpretations in the understanding of the selected risk statements. Using these statements as input artefacts, this kind of risk argument can be augmented with the recording of their risk interpretations that are worth exploiting in user debates. The risk of requirements that are not stable, for example, can be contextualized in terms of constant changes in requirement specifications in a given software project. For this kind of risk, for instance, a possible interpretation can be stated as changes in the business domain while the project is running. Here, development steps are the consultation of available sources of risk interpretations, the discussion of such interpretations with field experts, and the collection of risk interpretation state-

ments. The output artefact of this activity is this list of risk interpretations along with their selected risk statement.

*To analyze risk causes and effects:* it aims to assess the causal factors that may lead to the materialization of the risk in a software project. To do so, a cause-and-effect diagram called Ishikawa diagram [22] can be used as illustrated in Fig. 2 (A). This diagram contributes to the understanding of the risk nature, and the consequent representation of a scheme that captures such kind of problem. The input artefacts here are the constructed list of risk interpretations along with their selected risk statements. In this cause-and-effect analysis, risk interpretations can be taken as risk causes in the Ishikawa diagram. Interpretations can also be analyzed according to different software development contexts such as project, client and business, for instance. Here, development steps are the consideration of the identified risk as an effect in this diagram, the identification of causal factors related to the materialization of this risk, and the representation of these pieces of information in the cause-and-effect diagram modelling and refinement. The cause-and-effect diagram constructed is the output artefact of this activity.

*To reuse existing argumentation schemes:* it aims to identify reusable templates in scheme catalogues in the construction of a task-oriented scheme specification. In our project, the generalized formulation of the “argumentation scheme from cause to effect” [6] was used in the specification of the cause-and-effect kinds of requirement risk arguments. Although our schemes were motivated by this cause-and-effect template, other requirement risk schemes may also be constructed from other generalized argumentation patterns. For example, risk assessment argumentation schemes could be specified to support the analysis of key stakeholders’ claims as such arguments are understood via the “argumentation scheme from expert opinion” [6]. Here, development steps are the selection of argumentation scheme catalogues and consequent assessment of schemes available in them. In effect, this activity aims to find out generalized argumentation schemes, where their premises, conclusion and CQs are adjusted to reflect the debate needs of RM tasks. Here, the output artefacts are the formulations of generalized schemes to be reused in the specification of new argumentation templates for RM.

*To represent scheme premises, conclusion and critical questions:* it aims to deal with the concrete specification of the proposed argumentation scheme for RM. Here, the input artefacts are the selected risk statements, the risk interpretations identified, the results of the cause-and-effect analysis developed through the Ishikawa diagram and the generalized scheme specifications to be reused. In practice, scheme premises can be modelled from the causal risk statements analyzed in the cause-and-effect diagram. In effect, these premises can be described so as to capture risk interpretations. The conclusion of the proposed scheme is the very statement of the kind of risk that is the motivation for the scheme construction. Moreover, CQs can be formulated according to risk interpretations and the results of the cause-and-effect analysis. To structure a set of CQs, a knowledge engineer can exploit the argu-

ment nature of each question, checking if these questions capture exceptions to the structural rule of the scheme, for instance. In the “Argumentation scheme for risks of non-stable requirements”, for example, a question asks whether the projects’ application domain is really volatile, resulting in the constant changes in the specified requirements. Such questions show how debate users can attack the nature of the requirement risk arguments captured by this scheme, which is the true instability of the project requirements. In this activity, the output artefact is the concrete scheme specification, where this model is the object of additional steps of generalization, revision and validation.

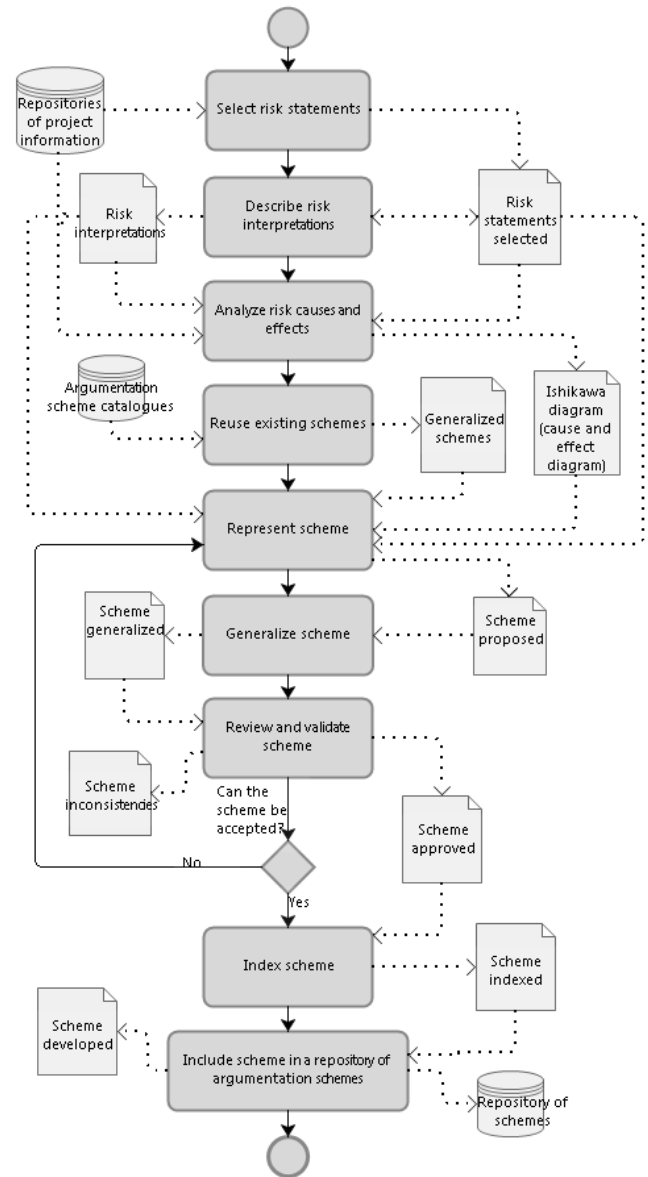


Figure 1. The process of argumentation-scheme development

*To generalize the scheme specification:* it aims to adapt the proposed scheme so that a more reusable task-oriented template specification is produced. Despite the general utility of generalized scheme formulations [6], risk-based schemes

ought to be specified in a language that regular stakeholders are more likely to cope with, promoting the understanding and reuse of these templates in collaborative debates. In addition to the rewriting of the terms and sentences used in the scheme description, this generalization activity is directed to the identification of scheme variables. These variables indicate places where these templates can be instantiated by debate participants. In debates, this is done when stakeholders use information from a current project situation in the construction of their scheme-based arguments. For instance, the first premise of the argumentation scheme for risks of non-stable requirements expresses that if the business domain is volatile, there will (might) have non-stable requirements. In the context of the collaborative debate of requirement risks in a particular project, this “business domain” term can be instantiated with the current project application domain. Here, the output artefact is the task-oriented scheme specification, which is now generalized to capture arguments for the RM problem.

*To review and validate the scheme specification:* it aims to evaluate the quality of the proposed scheme. To do so, a checklist is used in the revision and validation of this specification, which is the input artefact of this activity. In effect, this qualitative checklist should be adapted to knowledge engineers’ needs so as to approach the construction of schemes for supporting the development of selected RM tasks. Items that are detailed in such checklists assess the underlying understanding of the argumentation scheme specification achieved, as well as the kinds of risk arguments that the scheme intends to capture. Other aspects can also be reviewed are: if the risk proposal is stated in an objective way in the scheme conclusion, if the vocabulary used to describe the scheme elements is usual in the problem scenarios where these schemes ought to be exploited, if key CQs are listed in the scheme representation and if the level of detail used in the scheme specification is appropriate for typical stakeholders. The output artefacts of this activity are the proposed scheme along with a list of possible inconsistencies identified in these revision and validation tasks. Once such a checklist is completed a scheme specification can be approved due to the fact that this specification shows selected characteristics of quality. As modelled in our project through the exploitation of this development process, a resulting argumentation scheme for risks of non-stable requirements is specified as:

*Argumentation scheme for risks of non-stable requirements*

*Risk interpretations:* changes in the business domain during the project development; large number of change requests regarding requirements coming from clients; lack of criteria for managing requirement changes in the project;

*Major premise:* If the business domain (D) is volatile, there will (might) have non-stable requirements (R)

*Minor premise:* In project (P), the business domain (D) is volatile

*Conclusion:* There are non-stable requirements (R)

*Critical questions:* Is there evidence (X) that the high number of requests for changes in the requirement specifications is having a negative effect in project (P)? Are all requests for change (C) regarding requirement specifications (R) being accepted? Is the non-stability of requirements (R) due to the fact that the business domain (D) is changing frequently? To this scheme, the CQs, as in generalized formulations of argumentation schemes from cause to effect, are: How strong is the causal relation between the volatility of the business domain (D) and the instability of the project requirements (R)? Is the volatility of the business domain (D) the main (or single) cause for the instability of the project requirements (R)? Is there evidence (X) that the business domain (D) is volatile?

To the engineering of requirements, debate participants can question if the requirement engineering techniques are in place in the project, if there is a proper exploitation of these techniques, if the people involved in the development of requirement engineering tasks have the skills to adequately execute these tasks. The templates for these requirement engineering related CQs are: Are there requirement (elicitation, analysis, specification, validation, and management) techniques (T) to support (X) so that risk (R) is not in the project (P)? Are there requirement (elicitation, analysis, specification, validation, and management) techniques (T) being exploited properly by project stakeholders (K) so that risk (R) is not in the project (P)? Are the knowledge and experience of requirement engineers (E) adequate to do (T) so that risk (R) is not in the project (P)? In other words, it amounts to ask whether requirement engineers (E) have the right set of skills to develop the requirement engineering task (T). For instance, these requirement engineering questions in the context of the argumentation scheme for risks of non-stable requirements can be stated as: Are there requirement validation tasks (T) to support the exploitation of a volatile business domain (D) so that a risk of non-stable requirements is not in the project (P)? Are there requirement validation techniques (T) being exploited properly by project stakeholders (K) so that a risk of non-stable requirements is not in the project (P)? Are the knowledge and experience of requirement engineers (A) adequate to exploit a volatile business domain (D) so that a risk of non-stable requirements is not in the project (P)?

*To index the scheme:* it aims to organize the domain-oriented argumentation scheme proposed according to an indexing structure. This indexing consists of linking the scheme developed in a hierarchy of argument types. This argument-type concept aims to capture the nature of an argumentation scheme in an application problem, indicating what the scheme is about. Taking the approved argumentation scheme specification as input artefact, this specification is linked to a hierarchical list of argument types in this activity. For instance, the argumentation scheme for risks of non-stable requirements is an instance of *requirement risks*, which can also be understood as a type of argument for the *requirement risk management*. In essence, these indexing steps connect the new scheme specification to concepts used by users in the search of reusable schemes in argumentation repositories. Here, the output artefact is the indexed argumentation scheme.

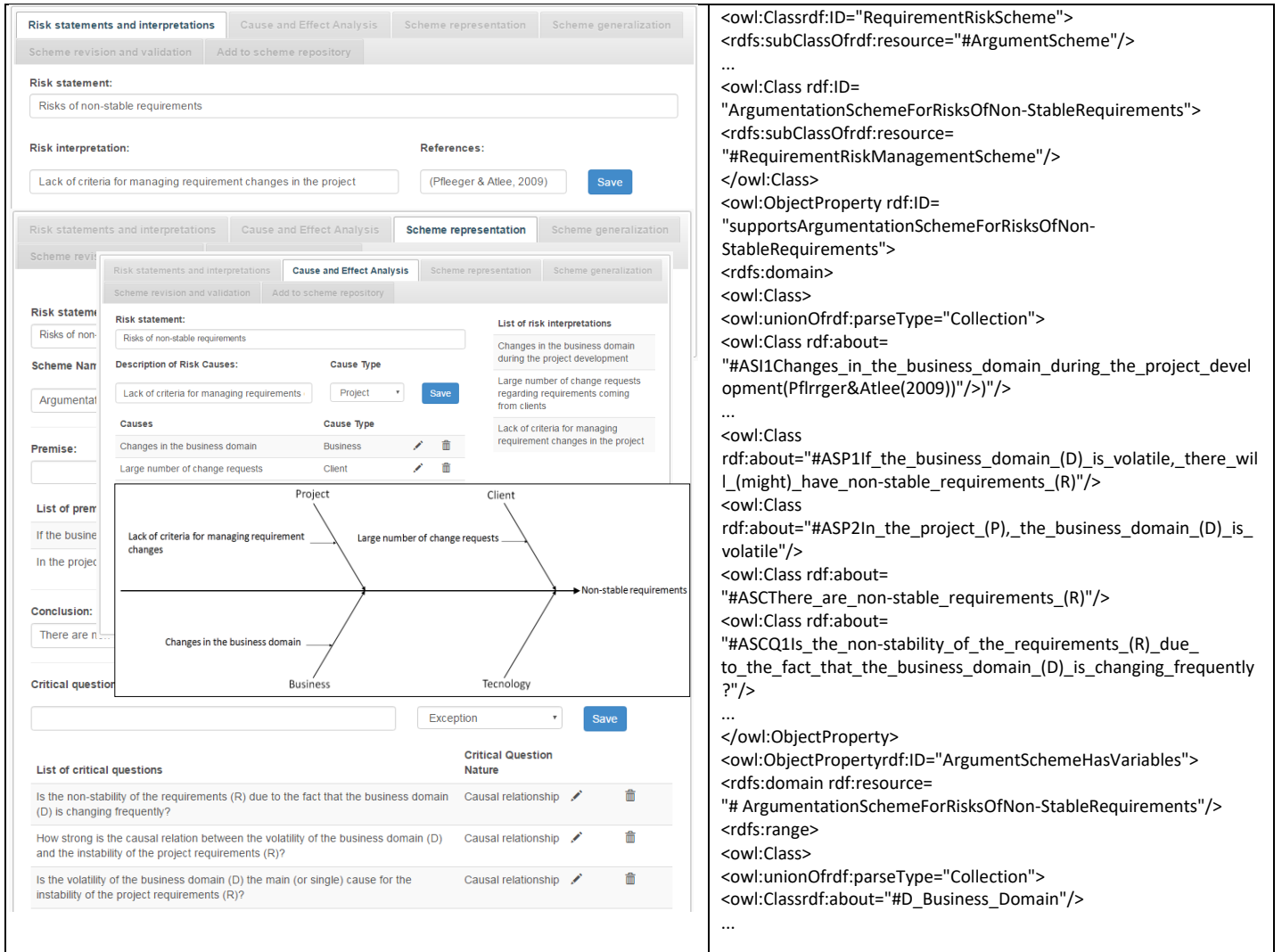


Figure 2. (A) A web-based system to support the execution of the scheme development activities proposed and (B) an excerpt of the argumentation scheme for risks of non-stable requirements represented in OWL

*To include the scheme in a repository:* it aims to construct a reusable argumentation repository, allowing the answer of queries and consequent reuse of developed argumentation schemes. The input artefact of this activity is the resulting scheme along with its indexing concepts. To compute inferences from scheme-based arguments stores in argumentation repositories, schemes need to be represented in computable formats of knowledge representation. As developed in our project, risk-related schemes are represented in the OWL format [23]. In this semi-structured model, scheme specifications can be queried by users and their computation tools. Fig. 2 (B) shows how the argumentation scheme for risks of non-stable requirements is represented in OWL. There, a class model is used in the capture of scheme interpretations, premises, conclusion and CQs. Scheme variables are also captured as classes, and relationships between scheme concepts are represented using class properties in this OWL model. In this final activity, the output artefact is the scheme developed which is represented in a semi-structured computational format.

#### IV. DISCUSSION

The definition of software development processes [24] allow stakeholders to standardize activities to be followed, the communication language to be used, and the techniques to be explored so that software development goals are achieved. To capture best practices, process models should not be immutable, since they need to evolve constantly to better support users in the solution of different problems. However, such improvements are possible provided that activities and artefacts that form these process structures are identified, allowing software engineers to evaluate and improve them continually. In this paper, we clearly identify a knowledge engineering process for the development of argumentation schemes for RM in software projects. Along with the discussion of each development activity identified there, the usefulness of this process is illustrated with the presentation of an argumentation scheme for risks of non-stable requirements (in addition to schemes presented in [9]), showing the reusable argumentation

artefacts that can be obtained when this development process is followed.

Most of the knowledge engineering work related to informal argumentation is focused on individual argument elements and their component parts that are identified in textual descriptions. This is usually developed through text-oriented annotation and diagramming resources adjusted to the analysis of argumentation concepts as implemented by the Araucaria system [15]. In this setting, generalized specifications of argumentation schemes [6] are also explored by users in the investigation of argumentation instances. As part of the development of critical argumentation skills, users are asked to recognize that an argument highlighted in such texts can be characterized by a given scheme. Although this line of argumentation work is relevant to the development activities proposed in this paper, these techniques are not directed or even organized as a knowledge engineering process for the specification of new instances of domain-specific argumentation schemes. Our process also considers the challenging linguistic and semantic analysis of large amount of textual information in specific domains, as shown by schemes for computer system safety engineering [7] and biological domains [8].

#### V. CONCLUDING REMARKS

Collaboration among project stakeholders in the engineering of requirements is crucial to achieve effective RM. The problem is that there is a significant gap between the information that is available in textual descriptions of RM users' arguments and the modelling and refinement steps resulting in generalized formulations of argumentation schemes.

In this paper, we approach the collection and structuring of well-formed stakeholders' arguments in collaborative debates of RM. In this context, this work contributes to the issue of laying out reusable argument templates to support these users in the construction of deeper analyzes of risks in their projects. In doing so, we detail a knowledge engineering process for the development of argumentation schemes for RM. We also describe a web-based system for supporting users on the development of the proposed scheme specification activities, aiming to facilitate the explicit representation of schemes in OWL. Preliminary evidence for the overall validity of this knowledge engineering process is demonstrated by a reusable set of argumentation schemes for the analysis of requirement risks in software projects as shown in [9].

Future work will involve attempting to exploit semi-structured approaches (e.g. OWL based) during the development scheme activities, rather than relying on informal scheme representations. We also plan to specify new scheme instances to be explored by project stakeholders in collaborative debates in different software project application domains. It amounts to involve both knowledge engineers and risk management experts to further experiment the proposed knowledge engineering process for the specification of schemes.

#### ACKNOWLEDGMENT

We thank the Brazilian Army for the financial support through the SIS-ASTROS Project (813782/2014), developed in the context of the PEE-ASTROS 2020.

#### REFERENCES

- [1] SEI, "CMMI® for Development, Version 1.3." p. 482.
- [2] F. Severo, L. M. Fontoura, and L. A. L. Silva, "A Dialogue Game Approach to Collaborative Risk Management" in The 25th Int. Conf. on Software Engineering and Knowledge Engineering, Boston, MA, 2013, pp. 548-551.
- [3] R. C. B. Pozzebon *et al.*, "Argumentation Schemes for the Reuse of Argumentation Information in Collaborative Risk Management," in Proc. of the 15th IEEE Int. Conf. on Information Reuse and Integration, Redwood City, CA, 2014, pp. 179-186.
- [4] C. Chesñevar, A. Maguitman, and R. P. Loui, "Logical Models of Argument," *ACM Computing Surveys*, vol. 32, no. 4, pp. 337-383, 2000.
- [5] P. Mccurney, and S. Parsons, "Dialogue Games for Agent Argumentation," in *Argumentation in Artificial Intelligence*, Boston, MA, 2009, pp. 261-280.
- [6] D. Walton, C. Reed, and F. Macagno, *Argumentation Schemes*: Cambridge University Press, 2008.
- [7] T. Yuan, and T. Kelly, "Argument Schemes in Computer System Safety Engineering," *Informal Logic*, vol. 31, no. 2, pp. 89-109, 2011.
- [8] N. L. Green, "Identifying Argumentation Schemes in Genetics Research Articles," in Proc. of the 2nd Workshop on Argumentation Mining, Denver, Colorado, 2015, pp. 12-21.
- [9] D. L. Siqueira *et al.*, "Argumentation Schemes for the Collaborative Debate of Requirement Risks in Software Projects," To Appear at The 29th Int. Conf. on Software Engineering & Knowledge Engineering (SEKE 2017), Pittsburgh, USA, 2017.
- [10] PMI, "PMBOK Guide: A guide to the Project Management Body of Knowledge," 2013.
- [11] K. Fenech, and C. De Raffaele, "Overcoming ICT Project Failures – a Practical Perspective," in World Congress on Computer and Information Technology (WCCIT), Sousse, Tunisia, 2013, pp. 1-6.
- [12] M. Warkentin *et al.*, "Analysis of Systems Development Project Risks: An Integrative Framework," *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, vol. 40, no. 2, pp. 8-27, 2009.
- [13] K. E. Wieggers, and J. Beatty, *Software Requirements*, 3<sup>rd</sup> ed.: Microsoft Press, 2013.
- [14] E. Bagheri, and F. Ensan, "Consolidating Multiple Requirement Specifications through Argumentation," in ACM Symposium on Applied Computing (SAC '11), TaiChung, Taiwan, 2011, pp. 659-666.
- [15] C. Reed, and G. Rowe, "Araucaria: Software for Argument Analysis, Diagramming and Representation," *Int. Journal on Artificial Intelligence Tools*, vol. 13, pp. 961-979, 2004.
- [16] C. Reed, D. Walton, and F. Macagno, "Argument Diagramming in Logic, Law and Artificial Intelligence," *The Knowledge Engineering Review*, vol. 22, no. 1, pp. 87-109, 2007.
- [17] S. E. Toulmin, *The Uses of Argument (Updated edition 2003)*, Cambridge: Cambridge University Press, 1958.
- [18] A. T. G. Schreiber *et al.*, *Knowledge Engineering and Management - The CommonKADS Methodology*, Cambridge: The MIT Press, 2000.
- [19] I. Rahwan *et al.*, "Representing and Classifying Arguments on the Semantic Web," *The Knowledge Engineering Review*, vol. 26, pp. 487-511, 2011.
- [20] I. Rahwan, F. Zablith, and C. Reed, "Laying the foundations for a World Wide Argument Web," *Artificial Intelligence*, vol. 171, no. 10-15, pp. 897-921, 2007.
- [21] B. Lawrence, K. Wieggers, and C. Ebert, "The top risk of requirements engineering," *Software, IEEE*, vol. 18, pp. 62-63, 2001.
- [22] K. Ishikawa, *Introduction to Quality Control*: Springer, 1989.
- [23] M. K. Smith, C. Welty, and D. McGuinness. "OWL Web Ontology Language Guide," <https://www.w3.org/TR/owl-guide/>.
- [24] M. Hull *et al.*, "Software development processes — an assessment," *Information and Software Technology*, vol. 44, no. 1, pp. 1-12, 2002.