Managing Risk Data: From Spreadsheets to Information Systems

José Barateiro
INESC-ID, LNEC
Information Systems Group
Lisbon, Portugal
Email: jbarateiro@lnec.pt

José Borbinha
INESC-ID
Information Systems Group
Lisbon, Portugal
Email: jlb@ist.utl.pt

***Abstract*—The goal of Risk Management is to define prevention and control mechanisms to address the risks attached to specific activities and valuable assets. Many Risk Management efforts operate in silos with narrowly focused, functionally driven, and disjointed activities. That fact leads to a fragmented view of risks, where each activity uses its own language, customs and metrics. That limits an organization-wide perception of risks, where interdependent risks are not anticipated, controlled or managed. The lack of integrated solutions to manage risk information, lead the experts to use spreadsheets as their main tool, impeding collaboration, communication and reuse of risk information. In order to address these issues, this paper presents a solution that integrates a Risk Management framework, including a XML-based Domain Specific Language for Risk Management. The proposed framework is supported by an information system to manage the definition or risks.**

I. Introduction

Risks always exist, whether or not they are detected or recognized by an organization. The ultimate goal of Risk Management (RM) is to define prevention and control mechanisms to address the risks attached to valuable assets [1].

Despite the fact that different communities use different terminology and phrasing to define risks, they share the main basic concepts. Indeed, a risk exists when a threat, that has potential to cause loss or harm, occur and is able to exploit a vulnerability/weakness associated with an asset that has a value to be protected. The type of assets depends on the organization nature, but might include physical entities (e.g., person, office), information entities and processes. When the vulnerability is exploited, it may cause an impact on the achievement of the organization's objectives.

Identifying, analyzing and modeling risks is one of the most critical tasks in the overall processes of RM. Traditional approaches, such as Fault Tree Analysis [2], Event Tree Analysis [2] and Failure Mode Effect and Criticality Analysis [3] are commonly used to model risks in the safety community. However, these approaches are not suitable to address the imminent risks that today's organizations face at multiple levels (both internally and externally).

Several models have been proposed to address risks at the organizational level, integrating the different views of the related stakeholders, such as the COSO Enterprise RM framework, KAOS [4] or GBRM [5]. Risks at the organizational level are covered by Enterprise Risk Management (ERM), which should be seen as an enabler to the organizations, providing risk information to executives and management boards, as well as audit committees, making it possible to incorporate this information to strategic and operational planning [6]. This increases the requirement to be able to exchange risk information, supporting its interoperability[1].

One of the main problems of RM is the fact that several RM efforts operate in silos with narrowly focused, functionally driven, and disjointed activities [8]. This leads to a fragmented view of risks, using its own language, customs and metrics. The lack of interconnection and holistic view of risks hampers an organization-wide view of risks, where interdependent risks are not anticipated, controlled or managed.

To support interoperability and provide a holistic view of risk information, moving from the traditional representation of risks in spreadsheets [9], to a controlled information system, this paper proposes a RM framework, including a XML-based Domain Specific Language for RM, and providing interoperability mechanisms to share information with related IT Governance and Enterprise Architecture activities.

The remainder of this paper is organized as follows. Section II describes our approach to address this problem, while section III details the developed solution. Finally, we conclude in section IV.

II. Approach

Identifying and controlling risks is a critical task in today's competitive world. Usually, this task takes place in the internal context of organizations. However, in complex scenarios, the problem comprises a whole spectrum of internal and external stakeholders, where risks can only be identified and adequately controlled through a proper knowledge of the overall enterprise design. This knowledge can be delivered by Enterprise Architecture (EA), which captures a vision of the "entire organization" [10].

Previous efforts build on top of the idea of extending with risk information, methodologies and models, commonly used to represent EA artifacts. For instance, in [11] the authors

[1]As defined by the Institute of Electrical and Electronics Engineers (IEEE), interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged [7].

978-1-4673-0784-0/12/$31.00 ©2012 IEEE

Table I
FORMALIZATION OF RM CONCEPTS

Concept	Formalization	Description
Attribute	$At(Name : string, Value : string)$	Descriptive properties identified by *Name* and quantified by *Value*.
Quantitative Domain	$D_{quant}(Type : string, Min : number, Max : number)$	Numerical range of *Type* ($\mathbb{N}, \mathbb{R}$) between the minimum *Min* and the maximum (*Max*).
Qualitative Domain	$D_{qual}(dq_1 : string, ..., dq_n : string)$	Limited list of admissible values $dq_1, ..., dq_n$.
Semi-quantitative Domain	$D_{s-quant}((dsq_1 : string, v_1 : \mathbb{R}), ..., (dsq_m : string, v_m : \mathbb{R}))$	Limited list of possible $(dsq_1, v_1), ..., (dsq_m, v_m)$ qualitative/quantitative pairs.
Domain	$D \equiv D_{quant} \cup D_{qual} \cup D_{s-quant}$	Can be a quantitative, qualitative or semi-quantitative domain.
Asset	$A(aName : string, aType : D_{qual}, aRef : D_{aRef}, a_1 : At, ..., a_n : At)$	D_{qual} determines the domain of asset types; D_{aRef} determines the reference of the asset to the EA.
Asset Value	$A_{Val} : A- > D$	Asset value (quantitative, semi-quantitative or qualitative).
Vulnerability	$V(name : string, vType : D_{qual}, asset : A)$	Identifies a vulnerability in an asset defined in A. An asset can have several vulnerabilities.
Vulnerability Exposure	$V_E : V- > D$	Function that determines the exposure of the vulnerability.
Event	$E(eName : string, eType : D_{qual})$	it can be a threat (bad event) or an opportunity (positive event).
Event Likelihood	$E_L : E- > D$	Initial estimation of the probability of occurrence of an event.
Risk	$R(rName : string, event : E, asset : A, vulnerability : V)$	Risk that occurs when an event is able to exploit an asset vulnerability. *rName* identifies the risk.
Risk Consequence	$R_C : R- > D$	Impact that occurs when event triggers the risk.
Risk Severity	$R_S : R- > D$	Severity of the risk (level of risk).
Block Control	$C_B : E- > D$	Control to block the event (reducing its probability).
Elimination Control	$C_E : V- > D$	Control to eliminate a vulnerability (reducing its exposure).
Reduction Control	$C_R : R- > D$	Control to reduce the severity of the impact produced by a risk.
Control	$C \equiv C_B \cup C_E \cup C_R$	Actions that can be taken to mitigate risks.
Cost	$Cost : C- > D$	Cost of implementing a control.
Policy	$P \equiv C_1, C_2, ..., C_n$	where $C_i \in C$

propose an extension to the Integrated Definition for Process Description Capture Method (IDEF) [12], while in [13] the authors propose an extension of the Business Process Modeling Notation (BPMN) [14] to include risk indicators. Another example is the creation of a UML profile to analyze system safety risks [15].

The approaches that extend previous models to integrate risk information can only be used with the underlying models, making it impossible to reuse this effort in scenarios that use distinct modeling notations, as well as limiting the EA representation to the semantics provided by the adopted models. This is not forcibly negative, as it provides an in-depth analysis in particular scenarios with a narrowly focused area.

In this paper, we intend to overpass these limitations and provide an integrated solution where risk activities can make use of knowledge provided by any EA representation. On the other hand, we also consider the requirement of being able to share information between distinct risk activities (that can be executed by distinct people, processes, systems, etc), making it possible to control risk in an integrated and holistic way.

The communication between risk and EA, as well as between distinct risk activities, where several models and representations are used, can only be achieved through a decoupled solution that supports the interoperability of the involved components. To support that, the solution proposed in this paper includes a formalization of the risk concepts, a domain specific language to support the instantiation of these concepts, and an architecture solution to support the integrated management of risk information.

III. SOLUTION OVERVIEW

Table I resumes the proposed RM concepts that are supported by *Risk-DL* [16], which is a *XML*[2] based vocabulary and schema. In fact, the *Risk-DL* defines the *XML Schema*[3], in the form of a *.xsd* file, that should be used to create *.xml* files defining risks. It uses the notation proposed in the relational model [17], where a **relation schema** describes the attributes of each concept, and a **relation instance** is composed by a set of instances of the concepts (tuples) defined in the relation schema. More formally, let $R(f_1 : D_1, ..., f_n : D_n)$ be a

[2]http://www.w3.org/XML
[3]http://www.w3.org/XML/Schema

relation schema, and for each $f_i, 1 \geq i \leq n$, let Dom_i be the set of values with the domain named Di. An instance of R is a set of tuples, where:

$$\langle f_1 : d_1, ..., f_n : d_n \rangle \, | d_1 \in Dom_1, ..., d_n \in Dom_n$$

Also, we define functions as $f : D- > R$, where f is the name of the function; D is the domain and R is the range of the function. Note that relations can be used to represent allowed domains or the range of functions.

The main purpose of using *XML* to model risks is to support the interoperability between distinct sources of risk information. Also, *XML* uses a human language that can be easily understood by people and computers, being highly portable and platform independent. Moreover, *XML* is an extensible language, which simplifies the evolution of *Rik-DL*, as well as the assurance of compatibility between different versions of the same language.

The main objectives of *Risk-DL* include, but are not limited to the following:

- support sharing, discovery, reuse and processing of the risk concepts;
- enable the alignment between risks and organization artifacts, by linking assets to records (e.g., business processes) managed within an organization EA;
- provide discovery and sharing of risk information between distinct stakeholders that are related to the same risk information, but can have distinct views and concerns with regard to those risks;
- facilitate feedback to improve and maintain risk information;
- reduce inconsistencies by formalizing the risk concepts and map them into EA artifacts;
- support an optimized definition of treatment plans, tracking controls to the related assets and reducing inconsistencies (e.g., controls that reduce a specific risk but may increase or add another one);
- provide an open specification that enables risk information to be categorized and support human-machine and machine-machine interoperability, either internally when different units produce risk information or externally across multiple organizations (e.g., external audit firms);
- produce a discoverable knowledge base where proven strategies to address common risks were already tested by previous treatment plans.

A. Reference Architecture

Figure 1 details the architecture proposed to support the integrated management of risk information. The *Operator* is responsible to interact with the system, providing a *Risk Description* that is transformed into the *Risk-DL Specification* of these risks, using the *Risk Modeling* component. The transformation into the *Risk-DL Specification* is supported by the Metadata Registry (*MDR*) component. This way, the architecture supports different versions of *Risk-DL*, as well as other risk representations.

The use of a MDR intends to ensure interoperability between different risk representations, as proposed by ISO/IEC 11179 [18], where an information system is responsible for managing and publishing descriptive information about resources (risk information). This way, the architecture supports different versions of *Risk-DL*, as well as other risk representations. The rational for this approach is based on the separation of concerns between the risk information and the services processing it.

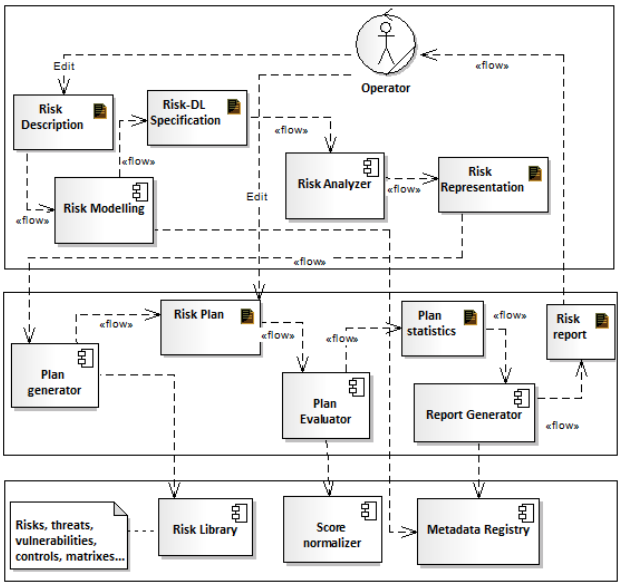

Figure 1. Architecture

The *Risk Analyzer* parses a *Risk-DL Specification* and generates an internal *Risk Representation* to be used and processed by the *Plan Generator*, which is responsible to produce options to manage risks (*Risk Plans)*, based on previous knowledge stored in the *Risk Library*.

The *Risk Library* represents a risk knowledge base, locally storing validated risk information as, for instance, risks used in previous scenarios, risk matrices, threats, vulnerabilities, assets, controls, plans, etc.

The *Plan Evaluator* produces a set of statistics that can be used to compare plans. When risks were defined according to different types of scores (quantitative, qualitative, semi-quantitative, or different scales), the *Risk normalizer* is responsible to normalize scores, turning it possible to compare and rank risks defined using different methods.

Finally, the *Report Generator* produces *Risk reports* to support the decision on the optimal plan to apply. Also, risk information must be delivered to different stakeholders (with different concerns), as proposed by the most prominent RM references. Having this in consideration, the *Report Generator* should provide different representations to view the risk information from the perspective of the concerns of every stakeholder.

B. System

The architecture proposed in the previous section can be instantiated in several ways, depending on design options. In the scope of this work, we developed a *web application* on top of the *Java* technologies and *JBoss Community*[4] solutions. This solution provides: an interface to model risks, importing and exporting of *Risk-DL*, a risk library, plan generator, and an interface to a MDR service.

On the other hand, the solution is highly extensible, since all the functions defined in table I can be extended through new *Java functions* (registering a *.jar* that respects the specified interface, or registering a *web-service* that receives a *Risk-DL* file and returns a new *Risk-DL* representation. For instance, a *web-service* to execute the function: Asset Value (A_{Val} : $A- > D$), receives the *Risk-DL* representation for a specific *Asset*, and returns a value in the domain *D* (note that domain validations are also performed inside the application).

The developed solution uses the *JBoss AS* application server, the *Seam Framework* as platform integrator, *Hibernate* to assure the persistence of the risk library and risk representation, and *Java Server Faces* to dynamically build the user interface.

Figure 2 shows the welcome screen of our solution.

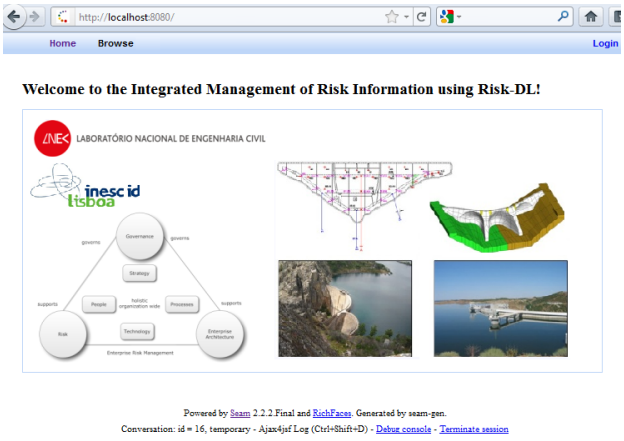

Figure 2. Welcome screen

IV. Conclusion

The risk community faces several challenges to model risk information. In fact, among other issues, risks involve a highly heterogeneous set of assets, events, methods, stakeholders and responsibilities, requiring adaptable methods and tools to support the exchange and interoperability of risk information. These issues are commonly addressed by the EA community, where organizations are modeled from the multiple views of the involved stakeholders (viewpoints). In this paper, we propose to take advantage of the EA to facilitate the exchange of risk information, as well as providing an organization-wide view of risks. We propose a solution that is decoupled from any risk or EA representation, so that it does not depend on any formalism, including a XML-based language to formalize risks, a reference architecture to develop solutions using this language, and a technical solution that implements the proposed reference architecture.

This work is being validated to model the risks associated to a dam safety information system. As future work, this approach will be further exploited in the scope of the TIMBUS project[5], where digital preservation risks (that can occur at several levels, e.g., hardware, software, processes) must be continuously monitored and assessed in a production environment that can involve multiple technologies.

[4]http://www.jboss.org

[5]http://timbusproject.net

Acknowledgment

This work was supported by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds and by the project TIMBUS, partially funded by the EU under the FP7 contract 269940.

References

[1] "Software Engineering Institute. Capability Mautiry Model Integration for Development. Version 1.3," Carnegie Mellon University, November 2010.

[2] M. Stamatelatos, W. Vesely, J. Dugan, J. Fragola, J. Minarick, and J. Railsback, "Fault tree handbook with aerospace applications," NASA, 2002.

[3] *DoD: Military Standard, Procedures for Performing a Failure Mode, Effects, and Critical Analysis (MIL-STD-1692A). US Department of Defense*, 1980.

[4] A. Dardenne, A. van Lamsweerde, and S. Fickas, "Goal-directed requirements acquisition," in *Science of Computer Programming*, vol. 20, 1993, pp. 3–50.

[5] A. Anton, "Goal-based requirements analysis," in *International Conference on Requirements Engineering*, IEEE Computer Society. Washington DDC, USA: IEEE Computer Society, 1996.

[6] S. Biazzo, "Process mapping techniques and organisational analysis: Lessons from sociotechnical system theory," *Business Process Management Journal*, vol. 8, no. 1, pp. 42–52, 2002.

[7] "IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries," New York, 1990.

[8] S. Maziol, "Risk management: Protect and maximize stakeholder value," Oracle Governance, Risk, and Compliance, White Paper, February 2009.

[9] M. Thoits, "Executive report on enterprise risk management technology solutions," Risk And Insurance Management Society, Tech. Rep., 2009.

[10] J. Schekkerman, *How to Survive in the Jungle of Enterprise Architecture Frameworks: Creating or Choosing an Enterprise Architecture Framework*. Trafford Publishing, 2006.

[11] J. Lambert, R. Jennings, and N. Joshi, "Integration of risk identification with business process models," *Systems Engineering*, vol. 9, pp. 187–198, October 2006.

[12] R. Mayer, C. Menzel, M. Painter, P. Witte, T. Blinn, and B. Perakath, "Information integration for concurrent engineering (iice): Idef3 process description capture method report," University Drive East College Station - Logistics Research Division, Wright-Patterson AFB, Tech. Rep., 1993.

[13] W. Cope, M. Kuster, D. Etzweiler, A. Deleris, and B. Ray, "Incorporating risk into business process models," *IBM Journal of Research and Development*, vol. 54, no. 3, pp. 4:1 –4:13, 2010.

[14] "Business process modeling notation, v1.1," 2008.

[15] B. Douglass, "Analyze system safety using UML within IBM Rational Rhapsody environment. Safety Analysis using UML," IBM, Tech. Rep., June 2009.

[16] J. Barateiro and J. Borbinha, "Integrated management of risk information," in *IEEE Federated Conference of Computer Science and Information Systems*, Szczecin, Poland, September 2011.

[17] R. Ramakrishnan and J. Gehrke, *Database Management Systems (Second Edition)*. McGRAW-HILL International Editions, 2000.

[18] "ISO/IEC 11179-1:2004. Information Technology – Metadata Registries (MDR) – Part 1: Framework," 2004.