A Context-Specific Definition of Risk for
Enterprise-Level Decision Making

Mirko Morandini¹, Alberto Siena¹, and Angelo Susi¹

Fondazione Bruno Kessler
I-38123 Trento – Italy
{morandini, siena, susi}@fbk.eu

Abstract. Enterprise-level decision making implies risk. In software companies, the choice of adopting an open source software component to be embedded into a commercial product may expose a whole business to risks arising due to this component. While general-purpose definitions of risk abound in the literature, in our work we seek an ontological definition of risk that allows us to understand how risks can relate the characteristics of a software component to the adopter’s business goals. In this paper we outline the challenges faced in building this ontological definition, and sketch some issues that are still open.

1 Introduction

The development of complex business information systems often relies on several tenths or hundreds of software components to provide better functionalities, reduce costs and time to market. Choosing the right components is a critical decision for a software company, as it could contribute to the success of the developed project, or determine its failure. Making the right decision requires to evaluate both the technical aspects of the components and their possible impact on the higher level business objectives. Component selection also implies the choice between Commercial Off-The-Shelf components (COTS) and Open Source Software (OSS) components. At a superficial look OSS components are software components that provide similar functionalities as COTS do, but are offered without a price. Assuming that the intrinsic qualities of the components can be evaluated with the same confidence degree for both OSS and COTS components, OSS components should apparently be preferable to COTS components, since they ease the possibility to achieve business objectives such as cost reduction, quality improvement, reputation and so on. However, missing a contract between the software company and the component provider, the company is exposed to other factors that are intrinsic to the fluid and open dynamic of OSS communities developing the component. Such factors are not fully perceivable at the time of the choice, but could appear in the future, causing additional costs. These factors represent risks for the company’s business objectives.

In the RISCOSS project¹ we are interested in (i) understanding the risks that lie in the selection of software components, and (ii) putting such risks in relation

¹ http://www.riscoss.eu
with the decision-maker’s business and its objectives. To do this, we made an effort to give an abstract notion of risk by means of an ontology, which relates risk to business goals. In the end we reached the conclusion that domain-specific concepts are necessary to provide semantics to the higher level concept or risk.

2 Conceptualising Risk in Business Domain

*Risk* is a concept of mind, such as *cost* or *value*. However, differently from value and cost, risk is characterised by the purpose of representing a *knowledge lack*, about the nature and circumstances (causes) of events, and about their effect on other business objects. So, differently form other business concepts, such as cost, value, or goal, representing “risk” as a single concept is difficult, because there is actually nothing to represent, and this raises some conceptual challenges.

**Risk vs. Risk event** A risk manifests as an event that reifies the risk in a certain situation. The event (seen at a class level) carries an important characteristic of the risk, the probability of occurrence. In natural language we often use an event name to name a risk. For example, “software project failure” can intuitively be understood as a risk for a software company, as it may generate various negative effects. However, it’s not difficult to argue that, e.g., if the software project is exploratory, or if the company has other better projects running, a software project failure can also be considered as a planned event with an accepted consequence and no significant impact on own goals; also, it could be considered an opportunity, if it allows to discover that a certain idea was not feasible. So, “software project failure” *per se* can not be considered a risk.

**Risk vs Risk impact** Another characteristic of the risk is its impact – the perceived severity of the consequences of an event on the business goals of the organisation. This concept represents a bridge between the abstract concept of risk and the more concrete concept of value in the organisation. It clarifies how a manifestation of the risk in the domain may concretely affect the domain itself. For example, “business disruption” can be conceived as a risk, although it represents the *impact* of an event’s occurrence on the business goals, for example of the project failure event. Such an impact may however be obtained from many different events. Often it may also be fully predictable or even desirable, not fitting any more the definition of risk.

**Risk vs. Risk category** This point relates a risk to a “class of risks” that is again related to the classification of the values/assets in the organisation. For example, concepts such as “security risk” or “operational risk” are mainly classifiers, since they are used to group together chains of causes, events or impacts on the basis of a common property. Obviously this raises a problem of classification, since the same event or impact may be considered as both a security or operational risk. However, this question seems to be merely a problem of granularity.
Risk as Relation The previous considerations drive to the idea of representing risk by enumerating its components. A risk becomes an n-ary relation to other concepts: the event, the impact, as well as the circumstances making the event possible, and possibly the actor having a goal and thus suffering from an impact which has a certain severity. A risk is moreover characterised by the derived value of risk exposure, obtained following risk analysis techniques and evaluated by combining event probability and impact severity, for example in a risk assessment matrix. Risk exposure can e.g. be low, medium or high. This solution, adopted in the RISCOSS project, will be described in the next section.

3 A Domain-characterised Risk Ontology

The RISCOSS Risk base ontology builds on the concepts of Event and Situation, as defined in the foundational ontology Dolce Lite Plus\(^2\) (see for example [4]) and also used in UFO [6]. It defines the central concepts of Risk, Risk Event, Goal, Activity, Cause, Consequence, Impact, and Measure, by positioning them in relationship with the surrounding concepts. It includes elements from the risk ontology defined in the EU project Musing [3], complemented with standards definitions such as ISO 31000, and terms defined in risk management methodologies such as CORAS [7], Cailliau and Lamsweerde’s goal-oriented risk analysis [2] and the \(i^*/Tropos\) goal-risk framework [1]. The concepts of measure and indicator were inspired by the terminology in [5]. The resulting ontology shares some base concepts and will finally be linked with the OFLOSSC ontology [8], which defines the main concepts for characterising Open Source and OSS ecosystems.

The base ontology was implemented in OWL using Protégé 4, complete with the necessary object properties and their inverses, with defined domains and ranges, property chains and property restrictions. Populated with individuals from the context of OSS adoption, it gives the possibility to classify them and to infer possible causes for a risk event.

Figure 1 displays the central concepts and relationships of the ontology\(^3\). The ontology aims to capture the components of risk in a specific domain – i.e., OSS component adoption. For assessing risks, we rely on objective measures available in the OSS domain. The class Measure and its refinement Indicator are essential for risk evaluation: a Measure instance is a numerical value of some property of a given OSS project, for example, lines of code, number of commits per day and so on. A combined Measure is an Indicator, if it is able to give an estimation of the likelihood of a RiskEvent, obtained in practice from statistical correlation or expert judgement.

On the other hand, a business Goal, satisfied by some (ideal) situation, has an importance, which is a numerical evaluation of the goal – may the importance be derived from internal business indicators or a subjective evaluation. The class

\(^2\) [www.w3.org/2001/sw/BestPractices/WNET/DLP3941_daml.html]

\(^3\) The complete set of concepts and object properties can be found in the Protégé OWL ontology accessible at [http://selab.fbk.eu/riscoss_ontology](http://selab.fbk.eu/riscoss_ontology)
Risk represents the n-ary relation among causes, events, and impacts on business goals.

4 Risk Identification

Along with the viewpoints previously presented, representing a risk will no longer be a single act, but a process, which consists in searching for the most atomic identifiable components of it. The process is necessarily domain-dependent, as it consists in analysing the domain to search for instances of possible causes, related events and impacts on business goals with their severity as perceived by the organisation. The corresponding risks emerge directly by relating those instances with each other. A Risk per se does not carry any additional information, considering that risk exposure is a derived value.

The domain analysis process allows to identify some instances of the classes Cause, Event and Impact. Table 1 shows examples of risk relations resulting following this definition. For example, a Project failure following a Community death in case of Low community activeness is different from a project failure following a community death due to Component obsolescence.

Notice that, in this way, the name of a risk actually loses its importance.

In practical usage and for common understanding, risks will be grouped according to different parameters, such as their cause, event or impact, creating de facto a risk category, typically named by the common cause or impact. Prominent examples in the OSS context are the security risk, including risks with impact on data confidence and data integrity, or the community risk, enclosing risks that has the behaviour of the community as cause.

5 Conclusion

A typical difficulty in the identification of risk is that it is almost always confused with either its causes, its consequences, or their impact on one’s goals. A
Table 1. Example of risk relations

<table>
<thead>
<tr>
<th>Cause</th>
<th>Event</th>
<th>Impact</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low activeness</td>
<td>Community death</td>
<td>Project failure</td>
<td>r1</td>
</tr>
<tr>
<td>Low activeness</td>
<td>Community death</td>
<td>Increased costs</td>
<td>r2</td>
</tr>
<tr>
<td>Low activeness</td>
<td>Quality decrease</td>
<td>Project failure</td>
<td>r3</td>
</tr>
<tr>
<td>Low activeness</td>
<td>Quality decrease</td>
<td>Increased costs</td>
<td>r4</td>
</tr>
<tr>
<td>Component Obsolescence</td>
<td>Community death</td>
<td>Project failure</td>
<td>r5</td>
</tr>
<tr>
<td>Component Obsolescence</td>
<td>Community death</td>
<td>Increased costs</td>
<td>r6</td>
</tr>
<tr>
<td>Component Obsolescence</td>
<td>Quality decrease</td>
<td>Project failure</td>
<td>r7</td>
</tr>
<tr>
<td>Component Obsolescence</td>
<td>Quality decrease</td>
<td>Increased costs</td>
<td>r8</td>
</tr>
</tbody>
</table>

conclusion of our work is that risk does not exist as a stand-alone concept. By representing risk as a relation, we are able to successfully classify a larger set of domain-specific risk types.

Several open issues have to be investigated in this conceptualisation of risk such as, for example, the distinctions and relationships between the concepts describing the objective reality (such as an event) and those that may describe the perception of the analyst (such as the impact and its severity).

Acknowledgement This work is a result of the RISCOSS project, funded by the EC 7th Framework Programme FP7/2007-2013, agreement number 318249.

References