Defining an Enterprise Architecture viewpoint to support Risk Management

Ruben Carlos Batista Louro

Thesis to obtain the Master of Science Degree in Information Systems and Computer Engineering

Supervisors: Prof. José Luís Brinquete Borbinha
Prof. Pedro Manuel Moreira Vaz Antunes de Sousa

Examination Committee

Chairperson: Prof. José Carlos Alves Pereira Monteiro
Supervisor: Prof. José Luís Brinquete Borbinha
Member of the Committee: Prof. Sérgio Luís Proença Duarte Guerreiro

October 2018
Acknowledgments

I would like to acknowledge and thank my dissertation supervisors Professor José Borbinha and Professor Pedro Sousa for the opportunity to work in this dissertation and for all the guidance, help and knowledge shared.

I also would like to express my gratitude to my family and friends who were always present to support me during the whole course of this dissertation, as well as for the continuous motivation and encouragement without which this important step of my life would not be achieved.

To each and every one of you – Thank you.
Abstract

Enterprise Architecture (EA) refers to an alignment between an organization's business processes, software and hardware, from which results an architecture description. Risk Management (RM), which is a process of dealing with risks associated with an organization's assets and finding solutions to prevent or mitigate such risks, can take advantage of these architecture descriptions in order to improve its efficiency. EA on the other hand, can also take advantage of the results produced by RM in order to support its architecture management. In order to combine the process of risk management with enterprise architectures, this aim of this dissertation is to implement a risk management scenario viewpoint in an existing enterprise architecture tool called Atlas. This project is therefore based in the study of good practices and standards that guide organizations through the risk management process and architecture development process.

Keywords

Risk, Risk Management, Enterprise Architecture, ISO Standards, TOGAF
Resumo

Arquitectura Empresarial (EA) refere-se ao alinhamento entre as camadas de informação, infraestrutura e negócio de uma organização, de onde resulta uma descrição de arquitectura. Gestão de Risco (RM), que consiste num processo de lidar com os riscos associados aos ativos de uma organização e encontrar soluções para prevenir ou mitigar esses riscos, pode tirar proveito destas descrições de arquitecturas para melhorar a sua eficiência. Por outro lado, EA pode tirar proveito dos resultados produzidos por RM de modo a suportar a gestão das suas arquitecturas.

De modo a combinar o processo de gestão de riscos com arquitecturas empresariais, o objectivo desta dissertação é implementar um cenário de gestão de risco, numa ferramenta existente de gestão de arquitecturas empresariais chamada Atlas. Este projecto irá portanto basear-se no estudo de boas práticas e normas que guiam as organizações no processo da gestão de risco e no processo de desenvolvimento de arquitecturas.

Palavras Chave

Risco, Gestão de Risco, Arquitectura Empresarial, normas ISO, TOGAF
# Contents

1 Introduction .............................................. 2
   1.1 The Problem ......................................... 3
   1.2 Objectives and Motivation ............................ 4
   1.3 Results .............................................. 4
   1.4 Organization of the Document ....................... 4

2 Related Work ......................................... 7
   2.1 Risk Management Concepts ............................ 9
      2.1.1 Risk Management Process ....................... 9
      2.1.2 Risk Assessment and Communication Techniques 12
      2.1.3 Risk Management tools overview ............... 15
         2.1.3.1 HoliRisk .................................. 16
   2.2 Enterprise Architecture Concepts .................. 17
      2.2.1 TOGAF Framework .............................. 18
      2.2.2 Archimate ...................................... 21

3 Atlas .................................................. 23
   3.1 Domain model ....................................... 25
   3.2 Use Cases .......................................... 28
   3.3 Architecture ....................................... 29

4 Problem Analysis and Requirements .................. 31
   4.1 Risk Management and Enterprise Architecture ....... 33
   4.2 Requirements ....................................... 34

5 Problem Solution ...................................... 37
   5.1 Impact, likelihood and risk level management ...... 39
   5.2 Configuration Category filter ....................... 39
   5.3 Filtered List ........................................ 40
      5.3.1 Navigation rules ............................... 41
   5.4 Probability / Consequence Matrix report ............ 41
6 Solution Demonstration

6.1 Pizza under Risk Scenario

6.2 Step-by-Step demonstration

6.2.1 Entity Management

6.2.2 Configuration Management

6.2.3 Probability / Consequence Matrix analysis

6.2.4 Filtered list analysis

6.2.5 Time Analysis
7 Conclusion and future work

7.1 Conclusions ................................................................. 67
7.2 Future Work ................................................................. 67

A Pizzeria Under Risk Scenario data ............................... 71
# List of Figures

2.1 Risk Management Process (Adapted from [1]) ............................................. 10
2.2 Risk Assessment Techniques from [2] ....................................................... 13
2.3 Consequence/probability matrix example ............................................. 14
2.4 Example of Check-list for Security Risk Assessment ............................... 15
2.5 Fishbone Diagram example .................................................................... 16
2.6 Example of a Risk Matrix report produced by HoliRisk (From [3]) ........... 17
2.8 TOGAF ADM ....................................................................................... 19
2.9 Archimate Architectural Framework (as seen in [5]) ............................... 21

3.1 Atlas domain model ............................................................................. 26
3.2 Atlas Use cases .................................................................................. 28
3.3 Atlas Architecture .............................................................................. 30

4.1 Pizzeria Under Risk Core domain model ........................................... 35

5.1 Form to add / edit a category configuration into the system ................... 40
5.2 Risk Filtered List showing information of a risk .................................. 41
5.3 Navigation rules established for the filtered list blueprints .................... 42
5.4 Probability / Consequence Matrix example configurations as input parameters .......................... 42
5.5 Probability / Consequence Matrix report added to Atlas ......................... 43
5.6 Probability / Consequence Matrix cell from the report added to Atlas .... 44
5.7 Lifecycle analysis between two dates on the control's list ......................... 44
5.8 User home page showing entry points .................................................. 47

6.1 Pizzeria Under Risk complete domain model ....................................... 51
6.2 Select Add Asset Form ........................................................................ 52
6.3 Press save after filling the form field ................................................... 52
6.4 Associate previous Asset to Objective .................................................. 53
6.5 Insert cause name .................................................................................. 53
6.6 Filling the qualitative scale field from the Consequence Scale Types ........ 54
6.7 Filling the quantitative scale with a value from the allowed range ............ 54
6.8 Assigning the new risk with an asset, a cause, a consequence and an event ... 55
6.9 Assigning the new control with a consequence and an event ................. 55
6.10 Select Add / Edit Configuration Form .................................................... 56
6.11 Valid options to use as filters for a configuration .................................... 56
6.12 Select the Batch Explorer tab ............................................................... 57
6.13 Run the selected batch ......................................................................... 57
6.14 Confirm the batch correct execution by seeing the green “ENDED” warning . 58
6.15 Select the Probability / Consequence Matrix from the Risk Visualization category .......................... 58
6.16 Select the previously created configuration to be used by the Probability / Consequence Matrix ................................................................. 58
6.17 Click the zoom in button ....................................................................... 59
6.18 Click the cell's configuration name to continue to the cell content .......... 59
6.19 Risks present on the selected cell of the Probability / Consequence Matrix ... 60
6.20 Select Objectives entry point from Global View category ....................... 60
6.21 Select the created objective from the objective list ............................... 61
6.22 Select the risk associated with this asset ............................................. 61
6.23 Risk Description filtered list ................................................................. 62
6.24 Select the details button for a specific control ..................................... 62
6.25 Fill each date of the Control’s states on the respective field ................. 63
6.26 Adjusting the time slider to make some of the previously created controls disappear .................................................. 63
6.27 Tick the lifecycle option ...................................................................... 64
6.28 Tick the gap option ............................................................................. 64

A.1 Pizzeria Under Risk Scenario data used in for the risk management scenario .... 72
List of Tables

2.1 Example of Quantitative and Qualitative scale for estimating the likelihood or financial negative impact .................................................. 12
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 The Problem</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Objectives and Motivation</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Results</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Organization of the Document</td>
<td>4</td>
</tr>
</tbody>
</table>
Every business emerges with the purpose of satisfying an existing need or idea. This purpose is formally defined, in the business organization, as the organization's objectives. In order to achieve its objectives a set of different assets have to be brought into the business, such as specialized workers, software technologies and hardware, which leads to a continuous increment over time in the complexity of the organization’s architecture. As a way of controlling this complexity, having a better and easier understanding of the organization’s structure and to facilitate business decisions, organization’s use tools for managing their enterprise architectures [6].

Aligned with the need for better management of their systems, for maximizing efficiency, effectiveness and minimizing losses when trying to achieve the organization’s objectives brings the necessity to take into account a proper way for handling with risks to those objectives [1]. In order to combine risk management and enterprise architecture, this dissertation will focus in the implementation of risk management capabilities into one of today’s tools for managing an organization’s enterprise architecture, called Atlas.

1.1 The Problem

Enterprise Architecture and Risk Management are two subjects which have different overall objectives. Enterprise Architecture is focused in the architecture of an organization and how to deal with its changes, while Risk Management is focused in identifying threats to the organization and how to deal with them. This project aims to analyze both subjects and find an alignment between them, in order to create a Risk Management scenario which aligns concepts from both parts. This scenario will be implemented into an enterprise architecture platform.

The platform which will be used is Atlas, which is a tool that was developed to support an efficient management of Enterprise Architectures, where any organization can configure it’s contents and analyze it through different types of customizable visualizations. It began as a Windows Desktop application and then evolved to a point that now supports a Web Application. One of the tool’s big capabilities is that it can be easily used to gather architectural information from other tools and use it for it’s own. The tool also provides an easy to use configurable navigation between it’s visualizations. Another of it’s big capabilities is that it allows to see how to see how the content evolves through time within a certain representation. This is achieved since the tool allows to manage information regarding the future state of the organization, this is, the organization’s To-Be state.
1.2 Objectives and Motivation

This project aims to understand how Enterprise Architecture and Risk Management can be aligned in order to create a scenario which is beneficial for both. For this, Atlas’s information management and visualization capabilities will be explored, in order to allow it to support risk management scenarios. This will include defining a risk viewpoint, where different ways of viewing risk related information, suitable for the stakeholders, will be defined in conjunction with the tool’s time travel capability. This will also result in adding to Atlas the ability to create multiple new reports with regard to risk, based in different types of risk assessment techniques, which will aid in the process of decision making when managing risks. The data that will be used as a risk management scenario during the development of this project will be from an example case study.

1.3 Results

The results which were achieved with this project were:

- Creation and population of the case study scenario within the tool.
- Setup of an automatic calculation for Risk Management formulas, to process and keep up to date values such as risk levels, impacts and likelihoods, taking into account the reduction amount of any existing controls.
- Filtered lists for analyzing an organization’s architecture.
- The option to create configurable configurations to filter the displayed information. Each of these configurations works as a set of preferences which can be used as an input for Filtered lists or the Probability / Consequence Matrix report, in order to create an adequate information filtering option.
- A Probability / Consequence Matrix report, which displays risks according to a risk assessment technique.
- The analysis of the scenario through time, which is possible by adjusting a time slider to see, for example, which controls are active at a certain time.

These will be further detailed in Chapter 5.

1.4 Organization of the Document

The rest of this report is divided into seven chapters.
Chapter 2 presents some related work by addressing the Risk Management component of the described problem, where fundamental concepts of Risk management are detailed, Risk Assessment Techniques to be used are described and some state-of-art tools are described. It also addresses the Enterprise Architecture component of the problem, where important concepts are described such as concepts that are used when describing an architecture and concepts used when using a common used architecture description language and TOGAF is also explored since it is the main standard for developing architectures. Chapter 3 is focused in exploring the Atlas platform capabilities and limitations, by introducing several points of view (Use cases, Architecture, Domain model) and providing information for how the problem can be addressed. Chapter 4 analysis the problem by exploring the alignment between Risk Management and Enterprise Architecture and defining the solution’s requirements. Chapter 5 describes information related to how the solution was achieved. It also details which technology was used and what was the software development process. Chapter 6 describes the example case study and then provides a step-by-step demonstration of the interaction with the Risk Management scenario. In the last chapter conclusions over this dissertation are taken and the future work discussed.
Related Work

Contents

2.1 Risk Management Concepts ................................................. 9
2.2 Enterprise Architecture Concepts ...................................... 17
This chapter describes the core concepts in Risk Management and Enterprise Architecture, and also examples of related relevant tools, in order to support a better comprehension of both the problem and the solution.

2.1 Risk Management Concepts

In order to provide a better interpretation of what Is Risk Management, it must first be understood the concept of Risk. At any given time, the activities executed by an organization to achieve its objectives are susceptible to occurrences which, due to their intervention, change those objectives reachability, for example by delaying its accomplishment. The prevention against the impact of such occurrences is therefore essential. For example, in a post office delivery service, a mailman needs his vehicle to be able to deliver packages. If this vehicle had a malfunction which made it unable to drive, having another vehicle as backup would be a measure of prevention, which would minimize the negative impact that would otherwise be inflicted on the operation of the delivery service. The uncertainty associated with the effect of occurrences over the accomplishment of an objective is what is defined as a Risk according to [1] and [7]. This definition supports that a risk is represented by the composition of a cause, an event and a consequence over an asset [7], which can be seen as the risk's properties. A cause is the uncertainty resulting from the change of the normal behavior of an artifact. An event is a situation that does not occur, as a result of at least one cause, which leads to an impact on the organization objectives. This impact can be described as one or more consequences. Using the previous post office delivery service example, the vehicle not being operational is an event, which can be caused due to the engine being broken, leading to the consequence of not being able to deliver any packages.

In order to attempt to control the source of risks and its outcome, organizations started to implement their own way of handling with risks. To support this need by organizations, the International organization for Standardization(ISO) released standard ISO 31000, which provides generic guidelines regarding Risk Management and also describes how the process of managing a risk should be executed, in a methodical and consistent way (as seen on Figure 2.1). Organizations that adopt their practice of Risk Management to be in accordance with this standard are capable of improving their efficiency through benefits such as improving controls, minimizing losses and improving the reachability of objectives [1]. Thus, any organization that wants to embrace risk management should follow this standard and so, it will be used to support the definition of Risk Management and related concepts on this dissertation.

2.1.1 Risk Management Process

ISO 31000 defines Risk Management as a set of "coordinated activities to direct and control an organization with regard to risk" [1]. This leads to the understanding that the overall intention when managing
risks is to control their severity, which can be done by implementing practices that are able to, for instance, minimize their likelihood or their consequences [7]. Such practices are defined as controls [7] and further on, it will be detailed how they appear.

![Risk Management Process](image)

Figure 2.1: Risk Management Process (Adapted from [1])

The process of Risk Management described in ISO 31000 [1], which can be seen as a whole in Figure 2.1, is composed of three main activities:

1. Establish the context
2. Risk Assessment
3. Risk Treatment

The first activity is to **Establish the context** where risks will be analyzed. This means defining a scope that includes external factors related to where the organization executes its activity and tries to achieve its objectives plus factors of the environment inside the organization itself, since every organization is different and the process of Risk Management should be adapted to each organization’s strategy.
and concerns [1]. This activity also includes the definition of the organization's risk criteria, which includes how the risk level and impact level are measured, and the level at which a risk can be accepted (Risk Acceptance level) [1].

The Risk Assessment activity concerns with identifying the risks, analyzing them and then evaluating if they should be treated or not. Therefore it can be split into three steps respectively and described according to ISO 31000 [1] as:

- **Risk Identification**: In this step, an event that has an impact in the organization's objectives, whether it is to enhance, degrade, delay or accelerate, is considered and, it's causes and consequences are identified. Thus, each risk is described according to a cause, an event and a consequence, and it should also be associated to a *risk owner*, which is a person or an entity, that becomes responsible and is authorized to manage that risk [7].

- **Risk Analysis**: An analysis is done over the information resulting from the previous step in order to determine what is the likelihood of the identified consequences, what is their impact and, from a combination of these, what is the risk level. When undergoing this analysis, existing controls have to be taken into account, consistency with the organization's risk criteria has to be maintained and factors that influence the impact and likelihood of consequences should be considered [1]. This analysis is expressed in a way that better suits it's processing, meaning that it's likelihoods and impacts can be qualitative or quantitative [7], or even a combination of these two [1]. When using qualitative values, the scale consists in categories, such as {Very High, High, Medium, Low and Very Low}. When using quantitative values, the scale consists of numeric values which can be for example {1,2,3,4,5}, where this example and the previous qualitative example both have five elements, and so, may correspond (as seen in Table 2.1), or a range from 1 to 10, where “1” and “2” can, for instance, correspond to “Very Low”.

- **Risk Evaluation**: An evaluation is done over the risks level against the organization's risk criteria, allowing for a better understanding of which risks have to be treated or if their existing controls over that risk are sufficient. This step also defines in which order the risks must be treated, depending on their severity.

The final activity in this standard's process for Risk Management is Risk Treatment. This activity consists mainly in modifying the risks [7] by applying at least one treatment to each of those risks. The treatments can include the following different types of actions according to ISO 31000 [1]: the risks can be accepted when the organization is willing to face it (*Risk Acceptance* [7]); the risks can be avoided by not executing activities and processes related to their causes (*Risk Avoidance*); the risks likelihoods and impacts can be minimized through the application of controls (*Risk Mitigation*); After a risk is treated, the
risk that remains is designed as residual risk [1], which should also be considered for further treatment or otherwise accepted.

Through all of these activities the communication and consultation with the stakeholders involved is maintained in order to have a better understanding of their perspectives, as well as helping them comprehend the need for certain actions resulting from this process [1]. This is the result of not all stakeholders having the same view over a risk [7].

Due to the unpredictability of risks and the constant change in the internal and external environment, a continuous monitoring and revision of each activity is needed in order to keep all of the information, regarding the process Risk Management, accurate and updated.

### 2.1.2 Risk Assessment and Communication Techniques

As previously mentioned, Risk Assessment is the combination of three types of actions which deal with the identification, analysis and evaluation of risks.

Although what happens each of this phases has been covered, it still has not yet been talked about how it can happen, this is, which methods can be used to perform them. ISO 31010 Standard was released as a way of addressing several tools and techniques than can be applied on each of them [2]. This standard not only describes how to apply each technique, as it also provides a comparison between the techniques in conjunction with guidelines for how to select which are the most suitable for each case. These can be consulted to their full extend in Figure 2.2, although only a few of them will be explored and therefore described, according to this [2] standard:

- Consequence/probability matrix
- Check-lists
- Cause-and-effect analysis
<table>
<thead>
<tr>
<th>Tools and techniques</th>
<th>Risk assessment process</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Risk identification</td>
<td>Risk analysis</td>
<td>Risk evaluation</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consequence</td>
<td>Probability</td>
<td>Level of risk</td>
<td></td>
</tr>
<tr>
<td>Brainstorming</td>
<td>SA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Structured or semi-structured</td>
<td>SA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>interviews</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delphi</td>
<td>SA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Check-lists</td>
<td>SA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Primary hazard analysis</td>
<td>SA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Hazard and operability studies</td>
<td>SA</td>
<td>SA</td>
<td>A&lt;sup&gt;3&lt;/sup&gt;</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>(HACOP)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hazard Analysis and Critical Control</td>
<td>SA</td>
<td>SA</td>
<td>NA</td>
<td>NA</td>
<td>SA</td>
</tr>
<tr>
<td>Points (HACCP)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Environmental risk assessment</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>Structure « What if? » (SWIFT)</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>Scenario analysis</td>
<td>SA</td>
<td>SA</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Business impact analysis</td>
<td>A</td>
<td>SA</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Root cause analysis</td>
<td>NA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>Failure mode effect analysis</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>Fault tree analysis</td>
<td>A</td>
<td>NA</td>
<td>SA</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Event tree analysis</td>
<td>A</td>
<td>SA</td>
<td>A</td>
<td>A</td>
<td>NA</td>
</tr>
<tr>
<td>Cause and consequence analysis</td>
<td>A</td>
<td>SA</td>
<td>SA</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Cause-and-effect analysis</td>
<td>SA</td>
<td>SA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Layer protection analysis (LOPA)</td>
<td>A</td>
<td>SA</td>
<td>A</td>
<td>A</td>
<td>NA</td>
</tr>
<tr>
<td>Decision tree</td>
<td>NA</td>
<td>SA</td>
<td>SA</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Human reliability analysis</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>A</td>
</tr>
<tr>
<td>Bow tie analysis</td>
<td>NA</td>
<td>A</td>
<td>SA</td>
<td>SA</td>
<td>A</td>
</tr>
<tr>
<td>Reliability centred maintenance</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>Sneak circuit analysis</td>
<td>A</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Markov analysis</td>
<td>A</td>
<td>SA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Monte Carlo simulation</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>SA</td>
</tr>
<tr>
<td>Bayesian statistics and Bayes Nets</td>
<td>NA</td>
<td>SA</td>
<td>NA</td>
<td>NA</td>
<td>SA</td>
</tr>
<tr>
<td>PF curves</td>
<td>A</td>
<td>SA</td>
<td>SA</td>
<td>A</td>
<td>SA</td>
</tr>
<tr>
<td>Risk indices</td>
<td>A</td>
<td>SA</td>
<td>SA</td>
<td>A</td>
<td>SA</td>
</tr>
<tr>
<td>Consequence/probability matrix</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>A</td>
</tr>
<tr>
<td>Cost/benefit analysis</td>
<td>A</td>
<td>SA</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Multi-criteria decision analysis</td>
<td>A</td>
<td>SA</td>
<td>A</td>
<td>SA</td>
<td>A</td>
</tr>
<tr>
<td>(MCDA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<sup>1</sup) Strongly applicable.
<sup>2</sup) Not applicable.
<sup>3</sup) Applicable.

**Figure 2.2:** Risk Assessment Techniques from [2]
**Consequence/probability matrix** is a technique that can be applied in all phases of Risk Assessment. This technique is used to calculate what is the risk level of risks, risk sources or treatments, as the result of combining the consequence and probability represented through a qualitative scale [2], which can be seen in Figure 2.3 1. From the positioning of risks in the different cells of the matrix, it can be interpreted which of them can be accepted (placed in the bottom-left of Figure 2.3), which have to be prioritized to receive a treatment and which require a deeper analysis. This is achieved since each cell of the matrix has a unique value assigned, that represents the risk level of risks in that cell. The consequence scale should be adapted to the types of consequence that are being considered, covering a range from the lowest to highest consequence possible, while keeping the probability scale unambiguous. In Figure 2.3, the categories can address, for instance, the financial loss, where an example that fits five categories was already given in Table 2.1, or property damage, where the category insignificant could represent minimal damage and would extend itself incrementally to the category severe, that could represent complete destruction of the property. The output of this technique is either the rating of each risk or a list with the risk’s rank and level [2].

![Figure 2.3: Consequence/probability matrix example](image)

**Check-lists** is a technique that can be applied in the Risk Identification phase. This technique is used to identify new hazards or risks, or to check if the existing controls are efficient. For this, it uses information from previous risk assessments such as hazards, risks, or controls that failed, to produce check-lists (see Figure 2.4 2) that will be applied to check if everything was covered by another technique, while or after it was applied [2]. This is achieved by having a person or team checking if the items of the check-list are in the system or not. The result of this technique can be a list of new risks or a list of

---


improper existing controls.

<table>
<thead>
<tr>
<th>Management</th>
<th>Yes</th>
<th>No</th>
<th>Action by Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Are attractive contents:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a) Kept in secure areas with alarm protection?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) Secured to desks, in the case of computers?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) Security marked.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is there a procedure to control the use of keys and a record kept of who has which keys?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cash:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a) Is there a cash control procedure?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) Is there a safe in use?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) Are safe keys removed from premises at night?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are the premises/grounds patrolled regularly?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do you have:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a) 24 hour guards?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) Security patrols outside normal hours?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) Resident caretaker?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.4: Example of Check-list for Security Risk Assessment

Cause-and-effect analysis is a technique that can be applied in the Risk Identification phase or for consequence analysis in the Risk Analysis phase. This technique is used to identify what causes may be associated to a certain effect, which may be an event or a problem, through a structured representation on a Fishbone Diagram of the possible causes that contribute to that effect, grouped in a set of categories. After the main categories of causes for an effect are defined as branches of the Fishbone Diagram, causes and sub-causes that fit those categories are explored by a team of experts and added to the respective branch or sub-branch. This allows for all of the possible factors to be considered, leading then to the identification of which of them are the more likely to be causes of the considered effect. An example of a Fishbone Diagram is presented in Figure 2.5 where the effect "Car will Not Start" is represented by a big arrow in the middle, this effect's cause categories are represented in the edges of the branches as "Materials", "Machines", "Measurement", "People" and "Environment" and the causes of each category are represented in the branches. The output of this technique is therefore a FishBone Diagram.

2.1.3 Risk Management tools overview

When talking about Risk Management tools, there two concepts that are important to consider: Risk Register and risk Report. A Risk Register is a document that has all risks and their details, resulting from the activities of the Risk Management Process, which include cause, likelihood, impact, control and owner [8]. A risk Report is considered any kind of report that a tool can produce as output or for analysis,

that brings knowledge over the existing information on the tool than concerns Risk Management, by using at least one type of Risk Assessment techniques.

On [3] some tools were explored by the author in terms of their risk management capabilities, where it is possible to see that the tools that address risk management tend to force the risk management process to be driven in a single way, pre-established by the organizations that developed the tools. The main analysis that these tools produce are based in Consequence/probability matrix or small variations of this technique.

2.1.3.1 HoliRisk

HoliRisk\footnote{http://holirisk.sysresearch.org/} is an web-based platform that supports all of the activities of the Risk Management Process (Figure 2.1), developed by INESC-ID. The tool's main operations for managing information are separated into Configuration and Population. While in Configuration, the tool allows to manage different domains, where in each domain a set of components and properties that it can use within certain context (for example inside an organization) are defined. In each domain, it is possible to configure a domain model in UML [3], where the relations between components and their properties are set. The properties can be either a basic type such as String or Float or it can be a range, which can be qualitative or quantitative. After the configuration is done, it is possible to create instances of the existing components in the Population operation, which consists in this tool's Risk Register. On [3], the author of this IST thesis added to the tool the capability of producing risk Reports of the type: Risk Matrix (seen in Figure 2.6), Filtered List and Fishbone. Fishbone and Risk Matrix (Consequence/Probability Matrix) were already covered in section 2.1.2. A Filtered List consists in a report that according to the author [3], resulted...
from an adaptation from Check-Lists technique (also covered in section 2.1.2), and provides a table with the list values from different components, which are associated by a relation.

Figure 2.6: Example of a Risk Matrix report produced by HoliRisk (From [3])

2.2 Enterprise Architecture Concepts

The evolution and expansion of organizations has led to an increase of complexity, for whom architecture is considered an important instrument of control [6]. An Architecture is defined as “the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution” according to [9] and [4]. In the context of this project it is important to introduce several notions used when describing an Architecture, which will be addressed in this section.

A Stakeholder is a person or organization [7] that intervenes in a system due to concerns (interests) such as needs, expectations, responsibilities, dependencies or requirements [4], and thus, that affects or is affected by the system [7]. The different types of concerns result from different types of Stakeholders involved in the system, each with their own capabilities of understanding the system’s architecture, thus creating the need to adjust an architecture’s representation to these, in order for it to be a proper communication tool. This need is fulfilled by Views [4], which are considered key artifacts when describing architectures by [10]. A View is therefore the representation of a perspective of the system’s architecture which addresses a set of concerns over this system ( [4], [10]). A View is established according to a set of rules imposed by a Viewpoint, that are dependent on the types of model that are being used to represent information. These models can be defined according to, for instance, BPMN or UML languages. A View can therefore be the result of combining aspects allowed by different models, meaning that it can incorporate one or more Architecture Models [4]. Types of relations to be used between elements, such as composition, dependency or constraint, are defined through Correspondences [4] within the
When the previously introduced Architecture concept is applied in organizations, the Enterprise Architecture concept emerges, which consists in the definition of an organization's structure, taking into consideration its objectives, in which, the main focus of the representation is the different components inside that organization, their properties and relations, resulting from an alignment between the organization's business processes, information systems and infrastructures [6]. It also defines how the architecture should further improve in terms of design and evolution [6].

![Conceptual model of an architecture description from ISO 42010](image)

Figure 2.7: Conceptual model of an architecture description from ISO 42010 [4]

### 2.2.1 TOGAF Framework

TOGAF is the most reliable standard for Enterprise Architecture [11] and it's main comprehensive source, resulting from an evolutionary development by The Open Group which started in 1995 and led to the most recent version 9.1, launched in 2011 [12].

The Open Group Architecture Framework (TOGAF) combines a detailed method and a set of tools
for supporting the development of enterprise architectures [12]. Before further describing the content of TOGAF, the different domains of architecture that are part of an enterprise architecture and whose development TOGAF supports [11], should be first understood. According to [12], there are four types of architecture: Business Architecture, which handles business strategy, governance, organization and key business processes; Data Architecture, which handles the structure of logical and physical data assets, within an organization, as well as assets regarding the management of data; Application Architecture, which handles applications that will be deployed and their relations to the business processes; Technology Architecture, which handles software and hardware that is required to support services in the previous architectures [12].

The Architecture Development Method (ADM) (seen in Figure 2.8) is the core component from TOGAF and consists in an iterative and cyclic process that describes how an enterprise architecture’s lifecycle should be developed and managed [12].

![Figure 2.8: TOGAF ADM](http://pubs.opengroup.org/architecture/togaf9-doc/arch/)

As Figure 2.8 shows, the ADM is a step-by-step method, where the steps correspond to different architecture phases that interact with each other, which cover the development of architecture domains [11]. Each of this phases has guidelines established on TOGAF [12], regarding the phase’s objectives, approach, inputs, steps and outputs.  

[5](http://pubs.opengroup.org/architecture/togaf9-doc/arch/), accessed 29, December, 2017
Besides the ADM method, TOGAF’s structure also includes a set of techniques and guidelines for using and adjusting the ADM to specific scenarios, the Architecture Capability Framework which provides a set of reference materials for establishing an architecture that deals with processes, roles, responsibilities and skills within an organization, the Architecture Content Framework which provides a structural model for addressing the structure and presentation of architectural content, and the Enterprise Continuum which provides methods for classifying architecture and solution artifacts, through a view over a repository of different related enterprise architectures, where the necessity of these architectures is the result of meeting the needs of different stakeholders [12].

TOGAF also has its own guidelines for how to handle risk management, in which it is defined what is risk management and what activities should be performed in this process, in the context of the development of enterprise architectures, where it is possible to see similarities with the definitions established by ISO 31000. According to TOGAF, risk management is “a technique used to mitigate risk when implementing an architecture project” [12], which is considered to be important since every architecture transformation always brings some kind of risk with it and this risk (or risks) should be identified, classified and mitigated before the beginning of the transformation process, in order to allow it to be tracked during the process [12]. Similar to ISO 31000, residual risks are also taken into account into their application of the risk management process, which consists of five activities:

1. Risk classification
2. Risk identification
3. Initial risk assessment
4. Risk mitigation and residual risk assessment
5. Risk monitoring

Risk classification involves classifying risks for purposes of delegating risk management and applying mitigation faster, where although the normal classification is time, cost and scope, it can also include other types of classification such as the architecture domain that the risk has impact on [12].

Risk identification is done as a result of maturity and transformation readiness assessments. Maturity assessment is done by using Capability Mature Models (CMMs) to identify what actions are needed to be able to achieve an architecture target state, where from the lack of capability to achieve that state, risks may be identified [12]. Transformation readiness assessment consists in determining if an organization is capable of accepting change, identifying problems that are impeding this capability and addressing them [12], where it is from these problems that risks may arise.

The Initial risk assessment consists in classifying the effect (impact), the frequency (likelihood) and the initial risk level of risks. This is done by using scales that organizations find to be more suitable for
them. These scales can have for example qualitative values, where the categories could be \{Negligible, Marginal, Critical, Catastrophic\} to measure the level of impact [12]. Thus, this activity is identical to the Risk Analysis activity that ISO 31000 defines.

In the **Risk mitigation and residual risk assessment** activity, controls to be used to mitigate risks are identified, planned and implemented with the objective of minimizing the risk level to an acceptable level, while prioritizing treatment on risks with higher severity first. The level of the impact and likelihood for the residual risk is then assessed to determine if the residual risk level is enough to be accepted or if other type of controls have to be applied [12]. Therefore this activity is similar to the combination of a part of the Risk Evaluation and the Risk Treatment activities from ISO 31000.

Finally, the **Risk monitoring** activity includes the core of the Risk Evaluation activity plus the monitoring aspect of the process described by ISO 31000. This activity consists in the acceptance of residual risks, by the IT (Information Technology) governance framework or, if necessary, by corporate governance, and then proceeding to monitor the implementation of controls, which is conducted in phase G (Implementation Governance) of the ADM method [12].

### 2.2.2 Archimate

Archimate is an architecture modeling language that is currently in version 3.0.1 and that is quite used for Enterprise Architecture modeling. In this language, the concepts that form a model are generic relations and elements, which are used to support the layered structure of the Archimate Framework [5], which can be seen in Figure 2.9. In each cell of this framework (Figure 2.9), it is possible to see a different stakeholder concern being handled.

![Figure 2.9: Archimate Architectural Framework (as seen in [5])](image)

The aspects group elements in function of their role on a behavior, where a behavioral element
can be, for instance, a process or a service and is represented in the Behavior aspect. If the element performs a certain behavior, for example a business actor or an application component, then it is represented in the Active Structure aspect. Lastly, the elements on which the behavior is performed on are represented in the Passive structure aspect. The exception to this is the Motivation aspect, where it’s elements represent the reasons that led to the necessity of the architecture [5].

The architectural layers represent different viewpoints and are aligned with the domains of architecture described in subsection 2.2.1, with the exception of the representation of information entities which is done by using objects of the Passive structure [5].

According to [5], the Business Layer has the business artifacts from an organization such as business processes and services that can be provided to users. The Application Layer has the information systems components, such as software applications that are used by the first layer. The Technology Layer has the infrastructure components that support the Application Layer. These can be for instance the type of storage, used to save information, or processing services, used to support the execution of a component, and are complemented by the equipments, materials or other physical resources from the Physical layer. The Strategy layer handles elements that represent what approach to take, by defining for example courses of action [5].

Implementation and Migration addresses change within an architecture, therefore being considered the most important layer to explore, since, as mentioned in the subsection 2.2.1, it is with architecture transformation that risks emerge. In this layer there are two Behavior elements, a work package and an implementation event. Work packages denote a set of actions with a specific objective, which have a defined start and end date. Implementation events represent a change in state related to implementation or migration [5]. The only Passive structure element is a deliverable, which is the result of work packages. The concept of Plateau, is used to refer to a specific state of architecture that lasts for a limited amount of time. It is an important concept for referencing the different architecture stages within each architecture development phase of the ADM method (phases B, C and D of Figure 2.8), since for each phase there is a current state (Baseline Architecture) and a desired state (Target Architecture), therefore being also useful for showing the transition between architectures [5]. The differences between two plateaus are described in a gap.
# 3 Atlas

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Domain model</td>
<td>25</td>
</tr>
<tr>
<td>3.2 Use Cases</td>
<td>28</td>
</tr>
<tr>
<td>3.3 Architecture</td>
<td>29</td>
</tr>
</tbody>
</table>
Atlas (previously known as EAMS) is an software tool for Enterprise Architecture modeling and it will be the tool that will be used to support the risk management scenarios. This chapter introduces the Atlas platform presenting its capabilities and features to support why it is a relevant and suitable tool to address the problem, while also exploring different points of view on the tool, its limitations and its alignment with risk management.

The tool was developed based in concepts that are used in Archimate so some similarities in its functionality will be noticeable with subsection 2.2.2. Atlas can be seen as an application separated into two main operations: configuration, in the Repository Manager and population, in the Explorer (similar to HoliRisk).

3.1 Domain model

The domain in which the information exists is called a repository. Multiple repositories can be managed in a tenant (as seen in Figure 3.1), which consists in a "physical database separation" [13]. For purposes of this dissertation problem, all configuration and data managing will be done in a tenant dedicated to research.

The Repository Manager is where all the configuration aspects are done. This includes the creation of different types of components of an organization’s domain model, as well as their property types. A component is called a class and the component’s properties are declared inside its class, similar to how they would be declared inside a class in a programming language such as Java (as seen in Figure 3.1). A property has a type which may be for instance text or numeric, or it can be defined as a bi-directional relation, to be used between two or more components. The definition of both the type and the relation is made as a basic type. Properties may have restrictions associated to them that either limit which type of class object can be inserted into that property, or specify a range of possible numeric values. Restrictions can be used to define the categories, whether they are qualitative or quantitative, that can be used to classify impacts, likelihoods or risk levels that would be represented by properties of a class.

Each instance of a class may have an owner assigned to it, which could be used to define the role of Risk Owner on instances what would represent risks. Alternatively, this could be done by creating a class that would represent risk owners and creating a relation to the class that represents risks.

It is also in this operation that different views over the architecture can be defined, which are designated as blueprints. These can have their layout, structure and content configured to adjust to different concerns of stakeholders within an organization and are specified by using queries that use a Atlas specific language. They can also be set to receive a component as a parameter, where in that case the blueprint will be applied to that specific scope. When viewing a blueprint after it’s generation, there is an adjustable time slider (also present in the instances of classes), from which is possible to see what
was the state of the content (As-Was), what is the content's current state (As-Is) and what is the planned future state (To-Be), which means that according to Archimate’s concepts, it is possible to see different Plateaus dynamically. In order for this to be possible, the classes of the instances to be shown in the blueprint must have these different states defined in their lifecycle tabs. This tab is present in all of the existing classes and it is used to configure which color should a component take when an instance assumes a certain state, in this case according to its time cycle where the state will change on specific dates, although it is possible to define other types of states, such as to view business contribution along time. For example, a non-existing component can be represented by a state which is not visible, a component under development can be shown in an gray state, an existing component in an green state and a component that expired in an red state. Thus making it only viewable after the slider covers a date that is on either of the last three states. Similar to a business contribution, a risk contribution could be defined, where it would be possible to see the impact on risks when inserting a new component or the impact on components of inserting a new risk.

Figure 3.1: Atlas domain model
Charts are also configured by using queries, and there are currently 12 types [13] which include bar, bubble, line, pie and scatter charts. Atlas also allows for reports to be configured in templates, which can show data, blueprint and chart information, either in Word or PowerPoint files [13]. Although there are many different types of representing data in a rich way, when considering to use risk management assessment techniques or variations of these in risk management scenarios, then Atlas's ideal visual representation to use is blueprints, since these represent objects, instead of just numbers or text, allowing a better analysis of the data, its properties and relations.

Atlas also allows for queries to be called over the existing components in such way that a user could, for example, ask to see all the instances that are related to a component by a certain relation. These can be configured and saved to be later used or can be written directly in the Explorer, to be used in that moment.

The Explorer is composed of different types of information explorers, which typically present a list of the data configurations in a panel. When accessing the Explorer the first thing that is show are the current profile’s entry points, which have the purpose of being accelerators for accessing, for instance blueprints or charts that are in the interest of a particular stakeholder. An example of their utility, aligned with the risk management scenario, would be to define an entry point for risk owners that would take them to the risks they are accountable for.

The Data Explorer is where, from a list of configured classes, the instances of each of these classes (objects) are managed, values are assigned to their properties and this information saved to the database. When assigning an object that does not exist to a property, the object will be created with the property’s type, meaning that objects can be created by reference. This way of managing data implies that whoever is doing it must have knowledge of how the metamodel is defined. A solution for a user that does not have this knowledge is to use forms, configured in the Repository Manager. A form is a simple way for a user to insert data into the repository, which automatically handles the complexity of managing classes and objects.

A list with the existing blueprints is also presented, through the Blueprint Explorer, where when one is selected, the respective view is generated and shown. From the interaction with the time slider within a blueprint, it is possible to perform either a gap analysis, where it is shown the evolution of components between two dates, a lifecycle analysis, where the components are colored according to their current state or a combination of both. The same occurs for charts, in the Chart Explorer and reports, in the Report Explorer. Lastly, the Explorer also contains the Scenario Manager. This is where edition of existing objects, removal of these or addition of new ones, in that repository, can be proposed by a normal user. This proposal must first be validated and approved, and only then can it be submitted to be published. Scenarios are useful for viewing the impact and behavior of the proposed changes upon the existing project, where queries can be used to test the results that would be produced by these changes.
Regarding user access, it is worth to note that only the type of users associated with each of the two main operations have the permission to edit those specific fields. An example of this is when a new user, by default, will only be able to create instances of already existing concepts and set their values, if it was allowed for him to access the repository that contains them. Another user type may be defined, such as the type “Manager”, which can be allowed to also configure components on a project’s metamodel. Permissions within repositories are also configurable, and the main idea behind them is that if a given object has no permissions, then everyone that can manage the object, but if a permission is set on that object establishing a person that could read it, then only that person would be allowed to read the object. The domain model from Atlas on Figure 3.1 further supports the relation of different entities on Atlas.

3.2 Use Cases

![Atlas Use cases](image)

All of the different actions that can be performed by a user in Atlas require specific permission, unless the user is a Administrator. As mentioned in section 3.1, Atlas operations are mainly to manage classes and their instances, inside a specific repository. Therefore, the responsibilities are split into three roles:
Repository Editor, Repository Manager and Repository Owner. The Repository Manager is the only one of the three roles that is able to create new repositories and create, read, update or delete (CRUD) classes and base types. He can also import existing configurations into repositories. When a repository is created, it is associated to a Repository Owner. Both the Repository Owner and the Repository Manager are capable of editing and deleting a repository, as well as managing (CRUD) Blueprints, Queries, Charts and Reports. They can also manage profiles permissions, either over the repository, allowing for users to see the repository and edit it, or over the four previously mentioned representations, being able to allow or not for another user to see and manage those repository’s Blueprints for example. The Repository Owner is also capable of managing navigations, which define what transition should occur when interacting with a representation. The Repository Editor is capable of managing all of the data instances, in a given repository. The responsibilities can be changed and in the default configuration, a "Manager" has the Repository Manager and Repository Editor permissions, the Administrator is the only one who can perform all of the operations and a User is a Repository Editor in the allowed repositories. The use case is represented in Fig. 3.2.

3.3 Architecture

Each different modeled project in Atlas is done in it’s own repository. The way the information in a repository is saved is through atomic transactions to Enterprise Architecture Database (EABD), whenever new objects are created or new configurations saved. EABD is therefore Atlas’s database, where other Atlas components connect to by an Hibernate Framework and, besides saving updated information regarding assets in a consistent way, it is also saves information from the asset’s life cycle. All of the other components of Atlas communicate between them through Rest Services (as seen in Figure 3.3).

Since Atlas is a tool that can be accessed through a web page, or installed as a windows application, there are separated components in the tool’s architecture to address each version. The Web application functionality is handled by EAMS Web in conjunction with EAMS Worker (Server). In one side, there is EAMS Web which is responsible for all the front-end to the user, separating Atlas’s main features into the Repository Manager, where all configurations can be inserted, namely basic types, classes, blueprints, reports, charts, queries and forms, and the Explorer, where all data objects can be managed based of the user’s permissions and the existing configurations used. In the other side, EAMS Worker is responsible for the back-end, providing services for generating blueprints, executing queries and batches, and loading scenarios on each of the repositories, in order to separate the work load [13]. A batch consists in a series of jobs, where a job is a configuration realizes a certain task, which can be importation, exportation, validation, execution or transformation of data. The way mathematical expressions are handled in Atlas is by sending data to a transformation job, which applies the desired formula and returns the
transformed result. The same can happen with a few other automatic operations, such as managing relations or property values.

The Windows application is handled by EAMS DeskTop and has the same purpose as the two previous components combined. Although both versions persist their data to EABD through EAMS Worker, the EAMS DeskTop can be configured to save its data locally, in XML format. In order to manage license validation, the versions are connected to EAMS License Server. The architecture can be seen in Fig. 3.3.

Figure 3.3: Atlas Architecture
Problem Analysis and Requirements

Contents

4.1 Risk Management and Enterprise Architecture ........................... 33
4.2 Requirements ................................................................. 34
This chapter presents an analysis of the problem, providing an alignment across Risk Management and Enterprise Architecture to establish the requirements for the problem of this dissertation.

4.1 Risk Management and Enterprise Architecture

As mentioned in section 2.2, an Enterprise Architecture consists in the representation of an organization’s structure, where all of the organization’s information is modeled through a set of components, properties and relations. This provides valuable information to improve the efficiency of the Risk Management Process, since it can analyze the architecture layers in order to identify risks. Since risks will be associated to assets which are part of the organization’s architecture, they will need to exist in the domain model. This implies that firstly, the domain model an organization to be supported, must be clearly defined in Atlas as classes, properties and relations to form its architecture with its specific domain. Secondly, the concepts that are handled through the process of risk management should be also modeled, if they are not already present, while having the appropriate relations to the architecture’s components, in order to support the risk management scenario. These concepts should include: objectives of the organization, assets, events, consequences, cause and controls.

From what we analyzed in subsection 2.1.1, having a risk associated to causes, events and a consequences, implies the existence of different likelihoods and impacts which combined will result in the risk’s level. Processing these values, across all existing relations to all risks, in an automated way, is essential to understand which risks have to be deal with and thus supporting the Risk Evaluation activity. The necessity of this automation is enhanced when controls are introduced as a mean of Risk Mitigation, which can modify the existing values of either likelihood or impact, resulting in a different risk level. Another concern is that if the values are all qualitative, or a combination of qualitative and quantitative, before proceeding to any calculation, the values of impacts, likelihoods and control reductions must be converted to a universal qualitative scale. Because Atlas is not capable of dealing with calculations and formulas directly, this feature will have to be implemented in a job, which will be executed externally.

The impact of architectural changes, resulting from the transformation from a plateau into another, was already identified as a source of risks in subsection 2.2.1. For example, introducing a new component in the architecture could raise new risks, which could have impact on other components. When introduced, this component could also be affected by another risk’s consequences, which would modify the impact value of those consequences. As stated by standard ISO 31000 (explored in subsection 2.1.1), one of the objectives of the risk monitoring and revision activity is to deal with changes in the environment and understanding whether they result in new risks or not [1]. This type of concern is also expressed in one of the steps from the Architectural Change Management phase [12] of TOGAF’s ADM (seen in Figure 2.8). In order to support both of them, it should be possible to analyze entity’s relations
through a set of *views*, to be able to determine which components they influence, overall improving risk identification efficiency. Atlas also provides a key feature for this type of analysis, with its *time slider on blueprints*, where it is possible to see the different *plateaus* of the organization, therefore further improving risk identification capabilities when used.

Since Enterprise Architecture by definition supports the existence of different types of stakeholders where not everyone shares the same view and knowledge of the architecture, the need to accommodate a user which is not familiar with the domain model of this risk management scenario is expected. Atlas provides the ideal solution for this problem through the use of *forms*, for managing *objects, relations* and the assignment of *property* value’s automatically. Thus, all of the domain model’s entities should have this management option.

A main concern of this dissertation’s problem is how to be able to provide valuable information from the domain model to the Risk Management process. This can be achieved through the use of reports based in one or more Risk Management techniques, such as the ones explored in subsection 2.1.2. Since this will be applied on an enterprise architecture tool such as Atlas, that was not developed to be able to provide risk specific reports over the information of the architecture on a repository, new types of report, that are configurable to adjust to each organization, need to be added to the tool.

Regarding the subject of inserting data in the risk management scenario, the concern of scalability must also be raised. What happens when, for example, instead of having a *blueprint (view)* dealing with about twenty risks and suddenly the scenario scales up, leading to that *blueprint* having to deal with two-hundred risks or more, is that it creates the necessity to categorize information, in order for it to be able to be filtered. This also allows for the creation of customized *views* that better suit the concerns of stakeholders. The filtering capability should therefore also be a feature of any report.

### 4.2 Requirements

A different set of requirements emerged from the need to incorporate Risk Management Capabilities combined with Enterprise Architecture on Atlas platform, in order to support the risk management scenario.

The architecture to be supported by the risk management scenario case study called Pizzeria Under Risk (which will be fully detailed in 6.1), whose key components can be seen in Figure 4.1. From analyzing this domain model it is possible to see that one single risk can be the result of multiple events and consequences. Since both events and consequences will influence the risk’s level, the solution must be prepared to process the components of each relation in order to find out the correct risk level. It is also possible to see that the consequences of the risks are related to one or more assets which belong to one or more objectives.
Requirements raised as important to have for each component (Atlas class) of the case scenario were forms, for risk registration purposes and lifecycle properties, for gap analysis with the time slider.

The requirements that were set by the stakeholders of Atlas platform regarding new types of reports that it should be able to produce as a consequence of data analysis in the models and that would support the risk management viewpoint are:

- **Risk Matrix**: Based in the Probability/Consequence matrix technique, where risks will be assigned to a specific cell in the matrix after their risk levels are calculated by a job.

- **Filtered List**: Based in the Check-Lists technique, where all information (instances or instance's properties) that is associated, by one or more selected relations, to a certain class instance, should be displayed by a Blueprint. Since one of the objectives of this technique is to understand if controls are efficient (detailed in subsection 2.1.2), it should also be possible to see the influence of controls in either consequences or events.

These types of reports would be used and managed by the risk owners of the organization, which should also have such information available from their own entry points when accessing the platform.

The requirements that were identified as calculations or automatic behaviors, which have to be created externally through jobs are:

- **Qualitative/Quantitative conversion**: Although the case study provides a qualitative scale (seen in Figure 4.1), a quantitative alternative is required for calculations, thus the combination of both types of value, while on the same context, also being required to manage.
• **Risk level calculation**: The process of calculating a risk level should be automatic and based on all of the risk’s events and consequences. This is required since as we can see in the case study on Figure 4.1, one risk can have more than one event or consequence.

• **Impact, likelihood and risk level after controls calculation**: The process of calculating a risk level on a risk with existing controls, should take into consideration its existing controls. All events or consequences that have one or more controls, must have their likelihood and impact level updated respectively. The change should then be propagated to each risk level.

• **Configuration Category filter**: When allowing for a user to operate within a set of chosen categories and being able to save such configurations to use at a later time or to use with a desired report, the filtering of the different set of options which the user may or may not choose to use has to be done in a job. The options provided to use as filters should be adequate to the case study and provide the possibility of filtering by key categories such as assets and objectives (based in the information seen in Figure 4.1).
5

Problem Solution

Contents

5.1 Impact, likelihood and risk level management ................................. 39
5.2 Configuration Category filter ....................................................... 39
5.3 Filtered List .............................................................................. 40
5.4 Probability / Consequence Matrix report ......................................... 41
5.5 Forms ...................................................................................... 43
5.6 Time slider .............................................................................. 44
5.7 Entry points ............................................................................. 45
5.8 Technology used ........................................................................ 47
5.9 Software Development Process ..................................................... 48
This chapter will describe the technology and development processes used to achieve the solution for the risk management scenario, as well as how the components of the solution were implemented in detail.

5.1 Impact, likelihood and risk level management

The capability to deal with the requirements Qualitative/Quantitative conversion, Risk level calculation and impact, likelihood and risk level after controls calculation, the solution combined this capability into a single job. Within this job, firstly the field which was filled by the user out of three possible value options (qualitative, with a scale of \{Very High, High, Medium, Low and Very Low\}, quantitative from a scale of \{5, 4, 3, 2, 1\} or a quantitative percentage scale from 0 to 100) to represent consequence’s impact, event’s likelihood and control’s reduction level is used to fill the other empty fields with the appropriate values. Secondly, each control associated to an event or consequence is taken into account in order to determine the final likelihood and impact respectively by using the values in range from 0 to 100 and applying the following formulas each time:

\[
\text{FinalLikelihood} = \text{Likelihood} - (\text{Likelihood} \times (\text{Reduction}/100))
\]

\[
\text{FinalImpact} = \text{Impact} - (\text{Impact} \times (\text{Reduction}/100))
\]

Finally, the risk level is calculated, taking into account all of the risk’s consequences and events, an by using the default formula of:

\[
\text{RiskLevel} = \text{Max} (\text{Impact}) \times \text{Max} (\text{Likelihood})
\]

These calculations are therefore applied to each event, consequence and risk from the case study introduced in section 4.2, while also following the calculation methods raised by that case study, aligned with the relations of the domain model seen in Figure 4.1. Both the starter value and current value of these three entities is kept in all of the three value options for analysis purposes, meaning that after these calculations, the percentage value is propagated to the other value options.

5.2 Configuration Category filter

In order to allow to filter and customize some view settings when using views like filtered-list or the Probability/Consequence Matrix, some additions were required to the domain model. A Risk Filter Configuration Manager class was introduced for purposes of containing access to all the other classes of the
domain model. This was a necessity since in case of a user not wanting to use any of the possible risk filter options (seen in fig. 5.1), then the filtering process would have to by default use all of the existing objects of such category. The only way to define a universal filtering job, required it to start by filling the missing fields, otherwise it would fail to accomplish its purpose.

![Risk Filter Configuration](image)

**Figure 5.1:** Form to add / edit a category configuration into the system

Associated to the Risk Filter Configuration Manager, a Risk Filter Configuration class was also introduced into the domain model, in order to contain information regarding that configuration preferences and probability / consequence matrix cells, in order to save the results of previously generated reports. The risk filter options were all of the existing entities in the domain model at the time, although if more categorization entities were to be added to the architecture, they would be easily added into the filtering calculations.

The job used for this effect works as follows: filter out configurations which do not require an update, filter out any entity which is not in the configurations preferences, fill any empty fields and filter everything down to risks.

### 5.3 Filtered List

In order to provide information through all of the relations of a certain instance, the following blueprints were defined based on the existing domain model:

- **Objective assets:** Shows all assets associated with a given objective.
- **Asset Description:** Shows all risks, objectives and consequences associated with a given asset.
- **Risk Description:** Shows the severity level before and after controls were applied and all of the controls, assets, causes, events and consequences associated with a given risk.
• **Cause description**: Shows all events and risks associated with a given cause.

• **Consequence description**: Shows the impact level before and after controls were applied and all of the assets, events and controls associated with a given consequence.

• **Control description**: Shows the reduction level, events controlled, consequences mitigated, and overall all the risks controlled by a given control.

These are accessed through any blueprint which was created to list all of each component of the architecture (it is also possible to use these filtered lists through the use of a desired configuration’s category options), by interacting with each other or by interacting with the Probability / Matrix report.

As an example, the Risk Description blueprint was provided in Figure 5.2. If the user wants to analyze the initial level and level after the controls were applied to the risk in a more precise way other that by looking at a qualitative scale, it is possible to inspect such risk's details and see both the starter impact and the current impact in quantitative scales.

![Figure 5.2: Risk Filtered List showing information of a risk](image)

### 5.3.1 Navigation rules

The navigation rules which were defined for the interaction upon double-clicking an object with and between the filtered lists are described in Figure 5.3.

### 5.4 Probability / Consequence Matrix report

This type of report was developed through the use of several blueprints, while taking into account the possibility of its content being restricted to a set of categories previously chosen in a configuration (as seen in Figure 5.4).

The Probability / Consequence Matrix report (seen in fig. 5.5) consists in a blueprint which shows the scales being used for likelihood and impact, and number of risks present in each cell in combination with the option to explore that cell's content. Due to scalability problems and the fixed size of each object represented in a blueprint, if the solution opted to show all the risks in each cell, such as a normal matrix.
Figure 5.3: Navigation rules established for the filtered list *blueprints*

Figure 5.4: Probability / Consequence Matrix example configurations as input parameters
of this type is usually found (in subsubsection 2.1.3.1, HoliRisk shows this representation in fig. 2.6), the matrix could become unreadable due to the huge amount of information that would be in each cell. The solution for this was to encapsulate the risks and instead show the number of risks which is present in that cell and allow for the user to analyze a specific cell at a time. Each cell required its own job, where it starts by checking which from all of the existing configurations requires an update and filtering the rest out, then it fetches that specific configuration’s risks and filters out the ones that do not belong in that cell, in order to find the number of risks.

The reason for each cell to have two objects, instead of just the object representing that cell’s risk number, is due to Atlas navigation rules between blueprints only being able to be applied to classes and demanding that the object whose properties will be shown is passed as an input. Since the cell risk number is an object which associated to a property of each cell and belongs the class of risk numbers, it was not possible to define navigation through the risks numbers, since there would be no way to specify to which cell to navigate to. Having the cells object represented, allows to navigate by drill-down to that specific cell’s blueprint by using a double-clicking. The interaction of that double-click will lead the user to the cell which contains its likelihood, impact and risks, showed by the blueprint in Figure 5.6.

This report required the addition to the domain-model of a cell specific class for each existing cell on the matrix, a full matrix class, a likelihood class and an impact class.

5.5 Forms

In order to avoid mistakes by a stakeholder which is not knowledgeable of the domain model, forms were established to propagate changes when adding, editing and deleting instances of the components
Figure 5.6: Probability / Consequence Matrix cell from the report added to Atlas

(objectsives, assets, causes, events, consequences, risks and controls) and configurations. An example of one of the created can be seen in fig. 5.1.

5.6 Time slider

For the gap analysis between the different plateaus, or on a single plateau to be possible, each entity of the domain model received four new lifecycle states: NoState (no color), Conceiving/Initial Estimation (gray color) State, Active State (green color) and Dead State (red color). The purpose of these states is to represent when a certain component does not exist yet, when it was estimated to be appear, when it became active and when it disappeared. This is used for example, to see when a certain control is expected to become active or when it is removed. In order to manage the dates of these states, field options for each state are presented when filling each existing form.

Figure 5.7: Lifecycle analysis between two dates on the control's list

On Figure 5.7 we can see the lifecycle analysis between two dates, on the existing controls, where we can see during this period of time: three controls that remained active (full green), three controls which went from the estimated state, into an active state and then disappeared (gray / red color) and
four controls which did not exist.

In order to help identify which events, consequences and risks have no controls, the possibility to perform a time analysis on these components with regard to controls was also added. In this type of analysis, the component's active control's states are directly shown without requiring to perform a drill-down.

5.7 Entry points

An entry point consists in a home page accelerator which allows to create a set of categories, each with its own options to navigate to. In order for the user to interact with the introduced features of this risk management scenario, a set of categorized entry points were defined as follows for the profile "Manager" (can be seen in Figure 5.8):

1. **Global View category**
   - 1.1 Global Combined View (combines all the others)
   - 1.2 Objectives
   - 1.3 Assets
   - 1.4 Risks
   - 1.5 Causes
   - 1.6 Events
   - 1.7 Consequences
   - 1.8 Controls

2. **Global View Filtered By Configuration category**
   - 2.1 Global Combined View (combines all the others)
   - 2.2 Objectives
   - 2.3 Assets
   - 2.4 Risks
   - 2.5 Causes
   - 2.6 Events
   - 2.7 Consequences
   - 2.8 Controls
3. Risk Visualization category
   3.1 Probability / Consequence Matrix
   3.2 Most Severe Consequences

4. Entity Management category
   4.1 Asset
      4.1.1 Add Asset form
      4.1.2 Delete Asset form
   4.2 Objective
      4.2.1 Add Objective form
      4.2.2 Delete Objective form
   4.3 Cause
      4.3.1 Add Cause form
      4.3.2 Delete Cause form
   4.4 Consequence
      4.4.1 Add Consequence form
      4.4.2 Delete Consequence form
   4.5 Event
      4.5.1 Add Event form
      4.5.2 Delete Event form
   4.6 Risk
      4.6.1 Add Risk form
      4.6.2 Delete Risk form
   4.7 Control
      4.7.1 Add Control form
      4.7.2 Delete Control form

5. Configuration Management category
   5.1 Add / Edit Configuration form
   5.2 Delete Configuration form

6. Risk Responsibility category
6.1 Risk Owners

6.2 User’s Risks

The first two categories have lists of elements of each entity and allow a user to navigate to the Filtered lists, with or without having filters. The third category allows access to the Probability / Consequence Matrix report. The fourth and fifth category deal with managing (add, edit or delete) data. The last category allows the user to see which risks he, or another user, is responsible for.

![Image of atlas application with Entry Points]

**Figure 5.8:** User home page showing entry points

5.8 Technology used

The technology that was used to achieve the solution with Atlas will be briefly described in this section.

Blueprints are specified through *queries*, which use a logic similar to *SQL* language but are represented in an *XML* format, making them a unique Atlas language.

Jobs used for data transformation are defined in *XML* and apply transformation rules specified in *CSV* format, editable in Excel sheets.

The operation of data importation from a different source uses Jobs defined in *XLST*, which allows the definition of scripts for changing the information format into the type of *XML* that Atlas reads.

The meta-model that is defined in a Repository is exported in *JSON*.

---

1 https://www.json.org/
5.9 Software Development Process

A "spiral" software development process approach will be followed in order to build the new functionality over the EAMS system and, therefore, to satisfy the requirements raised in section 4.2.

This process considers the client requirements and which of them are more crucial, in order to guide the course of the project through a set of cycles. These cycles will handle challenges such as which domain models to support, their population and the production of reports as a consequence of analysis to the data in these models, where for each cycle one or more of these challenges will need to be considered.

Each cycle starts by understanding what are the objectives of the client and establishing precise requirements. An analysis is then done to evaluate the best approaches that satisfy the requirements and a sketch of the solution is drawn. Afterwards, a prototype is developed and tested sufficiently to minimize its risks value. In the end of the cycle, the results are reviewed and then evaluated by the client, which will be the owner of any case study. Based on the solution's vulnerabilities, new requirements may be identified which lead to the planning of a new cycle, repeating itself until all of the client's requirements are met.

Each cycle was therefore be decomposed into four phases:

1. Objectives identification phase
2. Analysis and design phase
3. Implementation phase
4. Evaluation phase

To briefly describe how this solution was achieved, the case study's components were first modelled and populated in Atlas. This implied defining classes, relations and properties. The domain model was then further extended with the addition of the “Configuration Manager”, “Configuration” and twenty-five unique cell classes. The “Configuration Manager” would contain information about all of the existing elements in the system, in order to give the necessary information to each “Configuration” to use, according to that configuration’s filter options. Upon creation, each “Configuration” would also generate an object for each cell, which would hold information such as that cell’s risks and number of risks. To support the solution, three main Job’s were created: one for scale conversion and formula calculations, a second to process each configuration, and lastly a generic way of processing a cell’s risks and then calculating the number of risks, which was adapted to each of the cells. Forms managed all data manipulation, and would mainly create the objects and the necessary relations.
Solution Demonstration

Contents

6.1 Pizza under Risk Scenario .............................................. 51
6.2 Step-by-Step demonstration ........................................... 52
This chapter will describe a step by step interaction with the risk management scenario implemented as a solution. The architecture domain model used when developing the solution, as well as the data, belong to an example case study scenario which will also be described.

### 6.1 Pizza under Risk Scenario

This section describes the full Pizza Under Risk (PUR) scenario, which is a simple case study with low complexity used as a case study scenario for the risk management scenario developed as a solution for the problem.

The scenario represents how risk is managed in a restaurant where it deals with in-house service and home delivery service. The domain model (seen in Figure 6.1), although simple, is capable of capturing the core elements of the Risk Management process. It is possible to see that in addition to having a risk which can have multiple events and consequences (already analyzed in section 4.2), each of these can also have up to one control. The causes of an event are also represented in this version of the domain model, as an extend to Figure 4.1. The data used for this scenario can be consulted in Appendix A.

![Figure 6.1: Pizzeria Under Risk complete domain model](image-url)

The scenario represents how risk is managed in a restaurant where it deals with in-house service and home delivery service. The domain model (seen in Figure 6.1), although simple, is capable of capturing the core elements of the Risk Management process. It is possible to see that in addition to having a risk which can have multiple events and consequences (already analyzed in section 4.2), each of these can also have up to one control. The causes of an event are also represented in this version of the domain model, as an extend to Figure 4.1. The data used for this scenario can be consulted in Appendix A.
6.2 Step-by-Step demonstration

This section will demonstrate the steps required since logging in, in order to realize tasks such as creating components or analyzing information through a type of report. Each subsection represents a different task.

6.2.1 Entity Management

Scroll down to the Entity Management Category and click on "Add Asset" (shown in fig. 6.2).

![Figure 6.2: Select Add Asset Form](image)

6.2.1.1 Create an Asset

Fill in the form “Asset Name” field and click save (shown on Figure 6.3).

![Figure 6.3: Press save after filling the form field](image)
6.2.1.2 Create an Objective

Go back to Figure 6.2 and choose "Add Objective" instead. Fill in the form "Objective Name" field, associate the previous asset to this objective (shown in Figure 6.4) and click save.

![Figure 6.4: Associate previous Asset to Objective](image)

6.2.1.3 Create a Cause

Go back to Figure 6.2 and choose "Add Cause" instead. Fill in the form "Cause Name" field (shown in Figure 6.5) and click save.

![Figure 6.5: Insert cause name](image)

6.2.1.4 Create a Consequence

Go back to Figure 6.2 and choose "Add Consequence" instead. Fill in the form "Consequence Name" field, choose at least one scale type (shown in Figure 6.6) and associate this consequence with the asset created in step 6.2.1.1. Click save.

6.2.1.5 Create an Event

Go back to Figure 6.2 and choose "Add Event" instead. Fill in the form "Event Name" field, choose at least one scale type (shown in Figure 6.7) and associate this event with the cause created in step 6.2.1.3 and the consequence created in step 6.2.1.4. Click save.
Figure 6.6: Filling the qualitative scale field from the Consequence Scale Types

Figure 6.7: Filling the quantitative scale with a value from the allowed range
6.2.1.6 Create a Risk

Go back to Figure 6.2 and choose "Add Risk" instead. Fill in the form "Risk ID" field and associate this risk with the asset created in step 6.2.1.1, the cause created in step 6.2.1.3 and the consequence created in step 6.2.1.4 and the event created in step 6.2.1.5 (shown in Figure 6.8). Also assign a owner to that risk. Click save.

Figure 6.8: Assigning the new risk with an asset, a cause, a consequence and an event

6.2.1.7 Create two controls

Go back to Figure 6.2 and choose "Add Control" instead. Fill in the form "Control Name" field, the reduction amount and associate this control with the consequence created in step 6.2.1.4 and the event created in step 6.2.1.5 (shown in Figure 6.9). While on this form repeat this task in order to create a second control with a different reduction amount. This time it can be assigned with either just the consequence or event, if desired. Click save.

Figure 6.9: Assigning the new control with a consequence and an event
6.2.2 Configuration Management

Scroll down to the Manage Configurations Category and click on "Add / Edit Configuration" (shown in fig. 6.10).

![Figure 6.10: Select Add / Edit Configuration Form](image)

6.2.2.1 Select configuration options

Fill in the "Risk Filter Configuration Name" field and choose which options from the "Risk Filter Options" category (seen in Figure 6.11) should be used as filters for the displayed information. Any field that is left empty will consider that the user wants to use all available information regarding that field's category. Click Save.

![Figure 6.11: Valid options to use as filters for a configuration](image)

6.2.2.2 Select Batch Explorer

Navigate to the existing batches by clicking on the Batch Explorer tab, located on the bottom left side (seen in Figure 6.12).

56
6.2.2.3 Run the Risk Management batch

While on the Batch Explorer, select the "Risk Management" batch and press the run button (seen in Figure 6.13).

Figure 6.13: Run the selected batch

6.2.2.4 Wait for the Risk Management batch to execute

The “Risk Management” batch should take about 45 seconds to complete. Confirm this by seeing the green "ENDED" warning in the execution information (seen in Figure 6.14).

6.2.3 Probability / Consequence Matrix analysis

This task requires that the task of creating a configuration, described in 6.2.2, has been completed at least once. Scroll down to the Risk Visualization category and click on "Probability / Consequence Matrix" (shown in fig. 6.15).
Confirm the batch correct execution by seeing the green “ENDED” warning

Select the Probability / Consequence Matrix from the Risk Visualization category

### 6.2.3.1 Choose the configuration to be used as input

A popup window will appear listing all the existing configurations. The user should choose one of these configurations to be used as a source of information to be displayed by the Probability / Consequence Matrix. The possible configurations to select are either the ones created by the user (being selected in Figure 6.16) or a default configuration that shows all the information.

Select the previously created configuration to be used by the Probability / Consequence Matrix

### 6.2.3.2 Zoom in and move around

When the Probability / Consequence Matrix is shown, it will be zoomed out. Zoom in by using the zoom in button (seen in Figure 6.17) and scroll through the window to see each cell’s risk numbers.
6.2.3.3 Explore a cell

Since the configuration chosen on step 6.2.3.1 should be associated with the one risk created in step 6.2.1.6, find the cell that has a risk on it. The object on the right side of the risk number being displayed has the name of the configuration that is being used as a source of information. Press that object (seen in Figure 6.18) to explore that cell's content.

6.2.3.4 Analyze the cell’s risks

The list of risks of that cell are shown (seen in Figure 6.19). The user can continue to explore the risks by using the possible filtered lists discussed in section 5.3.
6.2.4 Filtered list analysis

Scroll to the Global View Category and click on "Objectives" (shown in fig. 6.20).

6.2.4.1 Analyze the objective

In this objective list we can see the objective created in step 6.2.1.2 (shown in fig. 6.21). Double-click that objective.

6.2.4.2 Analyze the risk

In this asset's filtered list (Asset Description) we can see the objectives, consequences and risks created previously. Double-click the risk created in step 6.2.1.6 (shown in fig. 6.22).
Figure 6.21: Select the created objective from the objective list

Figure 6.22: Select the risk associated with this asset
6.2.4.3 Risk related components

In this risk’s filtered list (Risk Description, seen in fig. 6.23) we can see the initial and current severity level of the risk as Very Low and the controls, assets, causes, events and consequences associated with this risk, which were created previously.

![Figure 6.23: Risk Description filtered list](image)

The user can continue to navigate through the objects if he wishes to explore further information by repeating the same process of the previous steps of double-clicking on objects, to explore the set of filtered lists described in 5.3 through the navigation rules which where described in 5.3.1.

6.2.5 Time Analysis

While on the *Data Explorer*, select the “Control” entity type and go to the details of the specific entities created in step 6.2.1.7, by clicking the buttons showed in Figure 6.24.

![Figure 6.24: Select the details button for a specific control](image)
6.2.5.1 Fill in the entity dates

When the properties open, select the "Time" tab and fill the available dates with the desired values (seen in Figure 6.25). Press Save and Close. Repeat this step if you created the second control in step 6.2.1.7.

![Figure 6.25: Fill each date of the Control's states on the respective field](image)

The user can then navigate to a filtered list which contains a list of controls by repeating task 6.2.4, but choosing Controls instead.

6.2.5.2 Adjust the time slider

When on a blueprint containing controls, the user can adjust the slider through time and see which controls are active in that date (seen in Figure 6.26).

![Figure 6.26: Adjusting the time slider to make some of the previously created controls disappear](image)
6.2.5.3 Lifecycle analysis

When pressing activating the lifecycle analysis (seen in Figure 6.27), the blueprint will show the state of each control in that date.

![Figure 6.27: Tick the lifecycle option](image)

6.2.5.4 Gap analysis between two dates

When pressing activating the gap analysis (seen in Figure 6.28), the blueprint will show the change along the two plateaus regarding controls. It is therefore possible to see which new controls were added, which were removed and which remained active.

![Figure 6.28: Tick the gap option](image)
7

Conclusion and future work

Contents

7.1 Conclusions ................................................................. 67
7.2 Future Work ............................................................... 67

65
This chapter presents this dissertation’s conclusions and provides some future work.

### 7.1 Conclusions

After studying how risk management should be done and it’s implications in the context of developing architectures, some alignments where found and explored in order to create a Risk Management scenario, while following guidelines for managing risk and developing architectures. For this to be achieved some limitations were encountered on the tool Atlas, that needed to be addressed. Ideas were taken from the exploration of a tool similar to Atlas that is focused only in risk management to support the lack of capability to produce analysis over the data of architectures according to a risk management viewpoint. The raised requirements are considered achieved. From these resulted the automation of processes such as formula calculations for risk management, configurable filtering options and ways of analyzing information such as a Probability / Consequence Matrix report.

### 7.2 Future Work

After the solution developed in dissertation, the possible future steps are:

- Allow for Atlas to be capable of managing single instances based on each of their properties instead of only allowing to process class instances as a whole. This limitation caused multiple problems when trying to manage relations of single instances, since its default process would modify all the instances of the same class as well.

- Allow for multiple dates to be assigned to a single state of a class’s life cycle. One of the ideas that was not possible to be explored due to this limitation was to have a category of states which would represent the for example the possible risk levels, thus allowing to analyze the visual change of all of the risks level through time while on a blueprint which listed risks. For this to be possible, we would have to consider cases where a risk starts in a certain state, changes to another state, but later comes back to a previous state and such feature is not possible.

- Use real case study scenarios with a greater degree of complexity.

- Allow for formulas to be a property value, in order to be possible to insert new different ways of calculating risk’s severity levels, or any other type of calculation, in a user friendly way.
Bibliography


Pizzeria Under Risk Scenario data
Figure A.1: Pizzeria Under Risk Scenario data used in for the risk management scenario