FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Integration of Reports for Enterprise Risk Management Processes in ERP Control

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Magister der Sozial- und Wirtschaftswissenschaften

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Christoph Rodler
Matrikelnummer 0125818

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer: Univ.-Prof. Mag. Dr. Walter S.A. Schwaiger
Mitwirkung: Univ.-Ass. Dr. Tatiana Miazhynskaia

Wien, 02.02.2011

(Unterschrift Verfasser) (Unterschrift Betreuer)

Integration of Reports for Enterprise Risk Management Processes in ERP Control

MASTER THESIS

to obtain the academic degree

Magister der Sozial- und Wirtschaftswissenschaften (Master of Science)

in the context of the study

Business Informatics

submitted by

Christoph Rodler
Matriculation number 0125818

at the
Faculty of Informatics of the Vienna University of Technology

Supervising
Supervisor: Univ.-Prof. Mag. Dr. Walter S.A. Schwaiger
Assistance: Univ.-Ass. Dr. Tatiana Miazhynskaia

Vienna, 02/02/2011 ________________________ ________________________

(Signature of Author) (Signature of Supervisor)

Christoph Rodler

Schulgasse 123/7
8224 Kaindorf
Austria

„Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.“

Wien, 02.02.2011 Christoph Rodler

"I hereby declare, that I have written this work independently, that I have completely indicated the sources and utilities I used and that I have clearly identified all borrowed parts in my work by a statement of the source - including tables, maps, and illustrations - where the text or meaning is taken from other works or the Internet."

Vienna, 02/02/2011 Christoph Rodler

Danksagung

Ich möchte all jenen danken, die mir während meiner Studienjahre zur Seite standen und mich auf so unterschiedliche Weise unterstützten, insbesondere aber …

… meinen Eltern Franz und Franziska für die Möglichkeit, das von mir angestrebte Ziel verfolgen zu können. Es ist eine unbeschreibliche Wohltat, ein solches Zuhause zu haben, woraus ich immer wieder neue Kraft und Motivation schöpfen kann.

… meinen Geschwistern Matthias und Marion für ihren engagierten Einsatz „wenn mal wieder der Hut brennt“, besonderes aber bei der fachlichen und sprachlichen Unterstützung von Matthias im Zuge der Fertigstellung dieser Diplomarbeit.

… meinen Freunden und Studienkollegen, welche ich in all den Jahren kennenlernen durfte. So lebendig wie ihre Charaktere, so abwechslungsreich sind auch die Erlebnisse. Sie gingen mit mir ein Stück des Weges und hinterließen dabei größere und kleinere Abdrücke. Jedoch haben sie alle eines gemeinsam: mithilfe ihrer Spuren konnte ich die Richtung beibehalten.

… Univ.-Prof. Mag. Dr. Walter S.A. Schwaiger für die Lebensweisheit, die Dinge aus verschiedenen Perspektiven zu betrachten, und für seine Bereitschaft, relevante Themen ausführlich zu diskutieren. Nicht nur an das Jahr als Studienassistent in seinem Bereich werde ich mich gerne erinnern, auch die anregenden Gespräche – nicht immer nur fachspezifisch – haben bei mir nachhaltig zur einem geschärften Gebrauch der Sprache beigetragen.

… meiner zukünftigen Frau Birgit, der ich hiermit auch diese Diplomarbeit widmen möchte. Danke für deine – des Öfteren – erprobte Geduld, deinen – mehrmals notwendigen – ermutigenden Zuspruch sowie deine Liebe und Güte in all den wunderbaren Jahren! Ich bin außerordentlich dankbar, dich an meiner Seite zu haben.

.

Abstract

Enterprise resource planning systems can be found in most enterprises and they are used to manage the resources. But the exchanging, holding and using of these resources is always done under uncertainty. The huge ERP systems in the market were developed decades ago, therefore new approaches, e.g. in risk management, cannot be efficiently integrated.

The term Business Informatics describes exactly the subject of this thesis: the introduction of concepts for enterprise risk management and their implementation in an ERP system. The establishment of enterprise risk management in an organisation shall assist to identify, assess and manage various risks, which may arise in the daily business. Two concrete risk models are discussed to illustrate the whole development process. The reliability model helps to estimate the probability of default of an equipment and the duration model evaluates a financial ratio, which enables to rate the risk of a financial instrument.

To put the introduced concepts into action a prototypical ERP system is the appropriate framework for the implementation of the risk models. The Department of Corporate Finance, Planning and Control as part of the Institute of Management Sciences at the Vienna University of Technology has developed a next-generation ERP system named ERPControl, in which the assessment and evaluation of both risk models have been incorporated. The outcome is the integration of risk management in ERPControl, which can assist a risk manager to create, reevaluate and monitor considered risks in an enterprise.

Kurzfassung

Es gibt heutzutage kaum ein Unternehmen, welches noch ohne den Einsatz eines ERP Systems arbeitet. Diese komplexen IT Systeme unterstützen bei der Verwaltung der Rohstoffe, Handelswaren und Investitionsgüter, welche vom Unternehmen verarbeitet, gehandelt oder benutzt werden. Die großen, am Markt befindlichen Systeme werden seit Jahrzehnten eingesetzt, dadurch ist aber die Systemerweiterung durch neue Ansätze, beispielsweise im Bereich des Risikomanagements, nur sehr ineffizient umzusetzen.

Der Begriff Wirtschaftsinformatik beschreibt den Kern dieser Arbeit sehr treffend: die Einführung von Konzepten für ein unternehmensweites Risikomanagement und dessen Integration in einem ERP System. Die Verwendung eines unternehmensweiten Risikomanagements unterstützt bei der Identifikation, Bewertung und Erkennung von verschiedenen Risiken, welche im Geschäftsalltag auftreten können. Anhand der Implementierung von zwei verschiedenen Risikomodellen wird der gesamte Entwicklungsprozess veranschaulicht. Mithilfe des Zuverlässigkeitsmodells kann die Ausfallswahrscheinlichkeit einer Maschine geschätzt werden. Das Durationsmodell liefert eine Finanzkennzahl, welche bei der Bewertung des Risikos einer Finanzanlage behilflich ist.

Die technische Umsetzung der Konzepte konnte mithilfe eines sich in Entwicklung befindlichem ERP Systems durchgeführt werden. Die Abteilung Finanzwirtschaft und Controlling vom Institut für Managementwissenschaften an der Technischen Universität in Wien entwickelte in den letzten Jahren einen dem aktuellen Forschungsstand entsprechenden Prototyp namens ERPControl. In diesem Softwareprojekt wurde die kontinuierliche Beobachtung von Risikoereignissen implementiert, was durch die fortlaufende Neubewertung aufgrund der Risikomodelle ermöglicht wird. Das Ergebnis ist die Integration von Risikomanagement in ERPControl, wodurch ein Risikomanager bei der Erstellung, Neubewertung und Beobachtung verschiedener Risiken in einem Unternehmen unterstützt wird.

Contents

1. Introduction

1.1. Overview

Most enterprises and organisations use various kinds of IT systems, which assist the manpower by organising and handling the daily business. The earliest IT systems were developed nearly half a century ago, the major IT systems, which are used today, were introduced in the business community in the 1980s. Therefore most of them suffer from the relicts of this time up to the present. Due to the economic and technological knowledge of that time, accounts and balances built the core components of those IT systems.

These days newly developed IT systems do not record their business activities as accounts, they focus on the exchanged resources and the according events. For that reason they are called enterprise resource planning (ERP) systems. The tasks of ERP systems are diversified: administrating stored resources, orchestrating workflows, recording payments, checking invoices, executing just-in-time production, providing interfaces to other IT systems like e-commerce or business-to-business collaborations, etc.

Enterprise risk management (ERM) deals with future potential risk events, which can have either a positive or a negative impact. Every business activity is risky, so an integrated ERM shall help to manage these risk events by reducing their impact and/or changing their probability of occurrence.

For the implementation of reports for ERM in an ERP system, ERPControl was chosen to put the concepts into action. The concepts to assess risks and to evaluate them are taught at the Department of Corporate Finance, Planning and Control as part of the Institute of Management Sciences at the Vienna University of Technology[1]. ERPControl, which can be described as a next-generation ERP system, is an ambitious software project of this department, which presents the theoretical concepts in an executable prototype. The actual standards for process management, economic resource exchange and IT technology have been implemented in ERPControl, so it matches best for the integration of risk management.

Document Structure

This master thesis is structured in three main parts. After a general overview the underlying economic concept of ERPControl will be introduced in Chapter 2. The following industrial standards are basis for the implementation, by what a broad acceptance of the outcome shall be secured in the community. The described management models assist to understand the

---

[1] Department´s homepage: http://www.imw.tuwien.ac.at/fc/

procedures for a reliable risk management. Finally ERPControl with its main implementation techniques will be presented.

Chapter 3 will illustrate the concepts for risk assessment and risk evaluation. Two different risk models will be explained in detail. The reliability model helps to estimate the probability of default of an equipment. The duration model evaluates a financial ratio, which enables the risk manager to rate the risk of a financial instrument.

Chapter 4 will present the implementation of the theoretical concept in ERPControl. The main realisation steps will be described: data persistence, business functionality, workflow processes and the resulting user interface.

1.2. Goals of Management Information Systems

Management Information Systems (MIS) are computer based information systems, which process information data for enterprises to support making decisions or monitoring special areas of interest. The focus of a MIS lies on the operational business management.

In the 1960s and 1970s because of the progress in computing the manager wanted to get more information out of their business data. So the first MIS were developed to extract information of the databases in run time.

To distinguish between the different types of MIS, Gorry and Scott-Morton developed a framework to classify MIS considering the type of problem and the managerial activity as depicted in Figure 1.

MIS, which operate on structured decisions, can process their input automatically without any input from a manager who has to make a decision. *“In a very structured situation, much if not all of the decision-making process can be automated.”*[1] So doing a short-term forecasting or an order entry, those are daily tasks and can be programmed easily. But when semi-structured and unstructured decisions are concerned, e.g. cash management or budget preparation, an algorithm to get a single result cannot be implemented in the same way. The authors named MIS for structured decisions *“Structured Decision Systems (SDS)”* and those, which handle problems below the dividing line in Figure 1, are called *“Decision Support Systems (DSS)”* [1].

A MIS in the meaning of DSS therefore shall extract information of the business data and support the manager to make the best decision. But to easily extract the right information or better, to get it at all, a well defined database structure according to the enterprise´s business is essential.

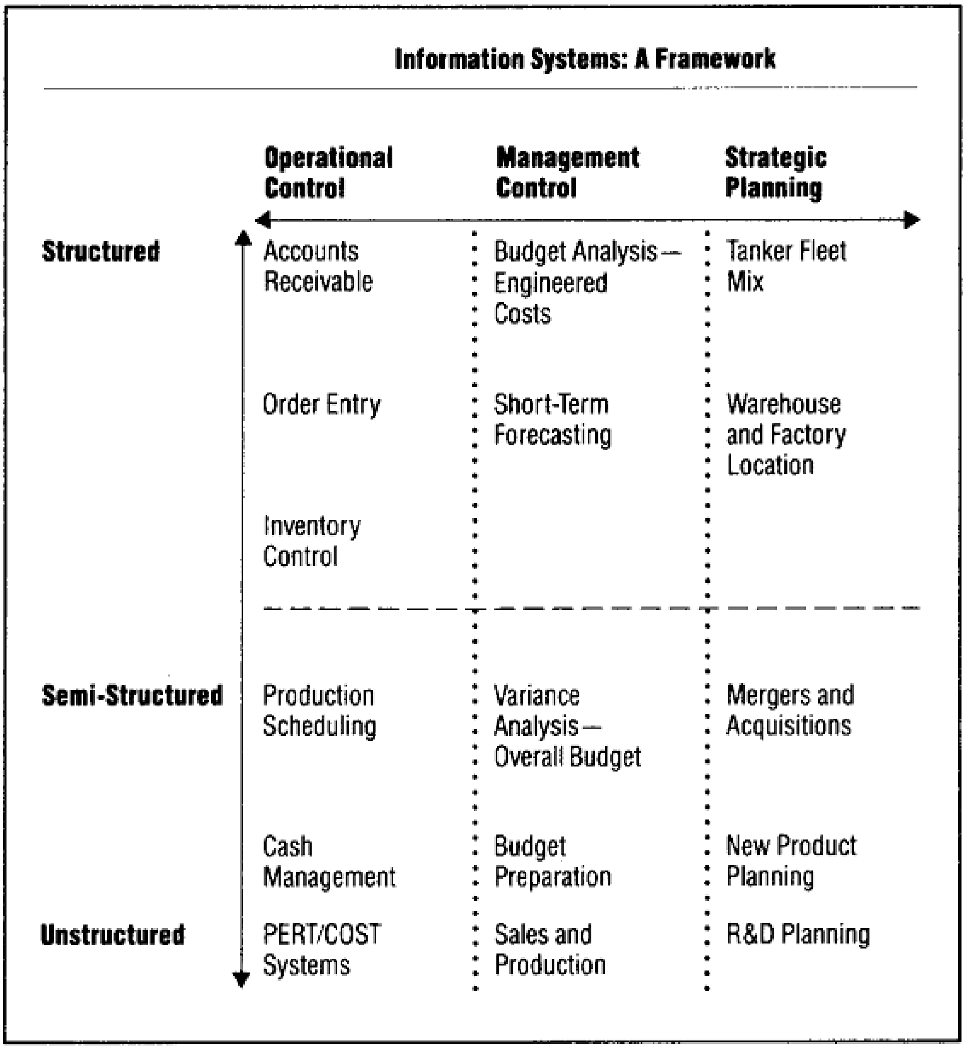


Figure 1: A Framework for classifying a MIS, developed in 1971 [1]

The business tools, which will be presented in the following, belong definitely to the semi-structured respectively to the unstructured decisions. An ERM of different resources with different risk models needs on the one side complex calculations, but on the other side a manager who interprets the results in the right way and who acts accordingly to their outcome.

1.3. Beginnings of REA-MIS

"Database design, therefore, is a process during which an attempt is made to mirror aspects of an identified reality (called the object system) in an abstract model (called the data model or schema)."[2]

First attempts to change the ideology of developing accounting systems were done by McCarthy. As an assistant professor of accounting McCarthy turned away from the double-bookkeeping method and established a new development method for ERP systems.

The starting point in McCarthy´s development process is the business analysis of the considered enterprise and afterwards the database design. *“The analysis process ... is one in which the database designer and accountant work together in (1) identifying the data requirements of different processes and decisions, (2) restructuring those data requirements in semantic terms using the REA model as an instance framework, and (3) combining the various restructured data specifications to form a conceptual schema.’*[2]

It is the goal to combine all *local views* to a *global data model*. The introduction of economic resources and economic events as parts of the Resource/Event/Agent (REA) model is used to avoid elements of the double-bookkeeping like e.g. accounts for credit and debit. The balance sheet and the income statement can simply be created by summing up the stored objects in the database.

The economic events illustrate the dynamic in the enterprise, in particular the increase and the decrease of resources, and they are used to create the income statement. The economic resources present the current situation in the enterprise and they are used to create the balance sheet. The resources also record the information, what is owned by the enterprise at a certain point in time.

The duality principle satisfies the economic rule, so that every increment causes a decrement of resources and converse. The definition of economic agents makes the need of accounts like receivable and payable unnecessary, because of the strong association between the three REA elements. Based on the research of McCarthy accounting in a REA based MIS is not any longer an independent and not-integrated system tool, but an incorporated part of the entire system.

2. REA integrated MIS

2.1. REA Ontology

“REA is for business systems what Einstein´s e=mc² is for physics.”[2]

This citation of Pavel Hruby describes in one sentence the deep impact of REA at business systems and their development. REA is in principle a simple approach, but with great possibilities for ERP systems.

Figure 2: Basic REA Ontology[3]

The first papers about REA were published in 1982. Initially it was developed to improve accounting systems and because of its simple, but clear economic concept, REA has been used in many areas like development, education and practical systems. It was the goal to reengineer the data structure of accounting systems so that it fits the requirements for modern enterprise resource planning systems [4].

Figure 2 presents the basic structure of the REA model. An economic event is associated on the one side with an economic resource, which can increase or decrease. On the other side there are the economic agents, which are the participations in a business transaction.

The idea behind REA is to rebuild the real business transactions in virtual objects, but without to use elements of the double-bookkeeping. Hruby introduced in [5] a software project, which illustrates the possibilities of a REA based ERP system. The balance sheet and the income statement can be computed by processing the data objects and summing them up. The stored resource flow is the journal, so if required every resource exchange can be reconstructed. With REA the economic principle that partners exchange goods can be

---

[2] Pavel Hruby, REA Technology, at the Vienna University of Technology in 2008

simply integrated into ERP systems that the system engineers can focus on other things than worrying about the differences between the real world and the development of the IT system.

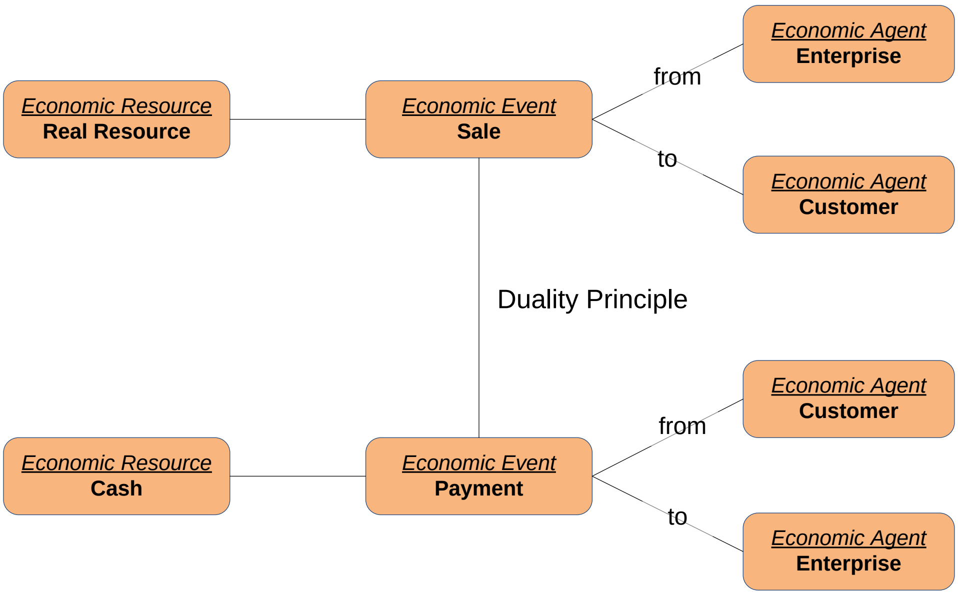


Figure 3: The exchange of resources designed in REA

In Figure 3 the exchange of resources is illustrated through a Unified Modelling Language (UML) diagram. Enterprise and the customer are the involved trading partners, which have the object type of economic agent. The first economic event is the sale of a real resource from the enterprise to the customer. The resource flow is from the agent enterprise to the agent customer. To satisfy the economic rule of the duality principle a second economic event completes the business transaction. The customer pays for the receipt of the real resource. The resource flow of the economic resource cash is from the agent customer to the agent enterprise. The two trading partners exchanged goods and a REA based ERP system records this business transaction without any complex mechanism.

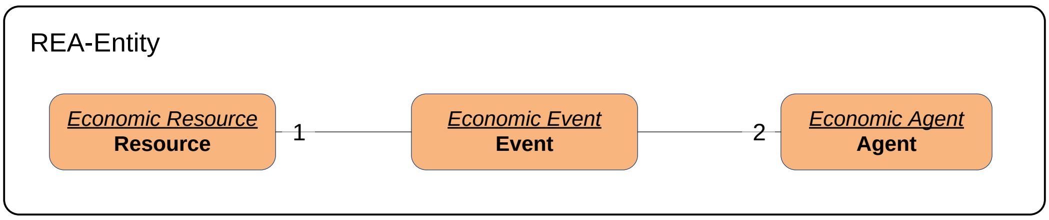


Figure 4: A REA-Entity with its components as UML diagram

The three parts of the REA model, the resources, the events and the agents, are associated with themselves; none of these elements can appear alone in a business transaction. There must be an agent, who owns a particular resource and who does something with it and a second agent who becomes the new owner. The "do something" is the economic event, which describes the resource flow in a more exact way. The combination of these three parts is called a REA entity as depicted in Figure 4, a comprehensive explanation can be found in[6].

2.1.1. REAv1 – recording past Events

If an agent sells a resource, it is an economic rule to get something back, e.g. money. This can be described by the duality principle: a second agent, the counterpart, returns a resource, which has the same value as the received resource. A REA entity on the one side causes a second REA entity, that in the end both agents exchanged resources of the same value. Another important fact is that the value of the two events has to be equal. This economic rule namely the value restriction has to be satisfied, even if more events are associated in one business transaction, e.g. if an invoice includes more than one line of item.

As shown in Figure 5 the business case has two important tasks: it combines the according REA entities to implement the duality principle and it verifies a satisfaction of the value restriction. So when a business transaction is completely concluded only correct changes will be recorded in the database. This ensures a consistent system state at every time.

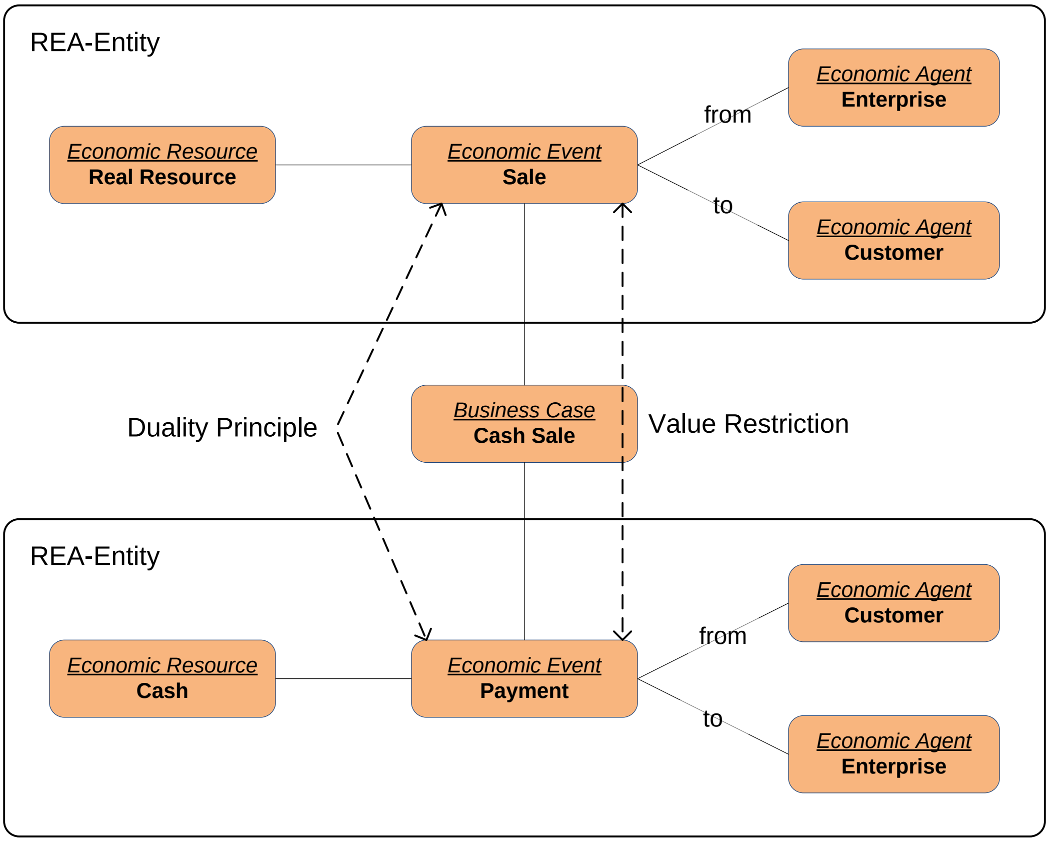


Figure 5: The REAv1 model with the two REA-Entities collected in one Business Case

2.1.2. REAv2 – handling future Commitments

Traditional accounting systems just record past events which have already happened. But modern ERP systems also have to handle future events. Things can happen in the future, e.g. an order, but they need not to become a business transaction. So the idea is to generate future events called economic commitments which can be fulfilled and stored as economic events in the database.

Another distinction between past and future events is that all, which will happen in the future, has a probability. Things can happen or not, the future economic commitments can take place in the future but only with a certain probability.

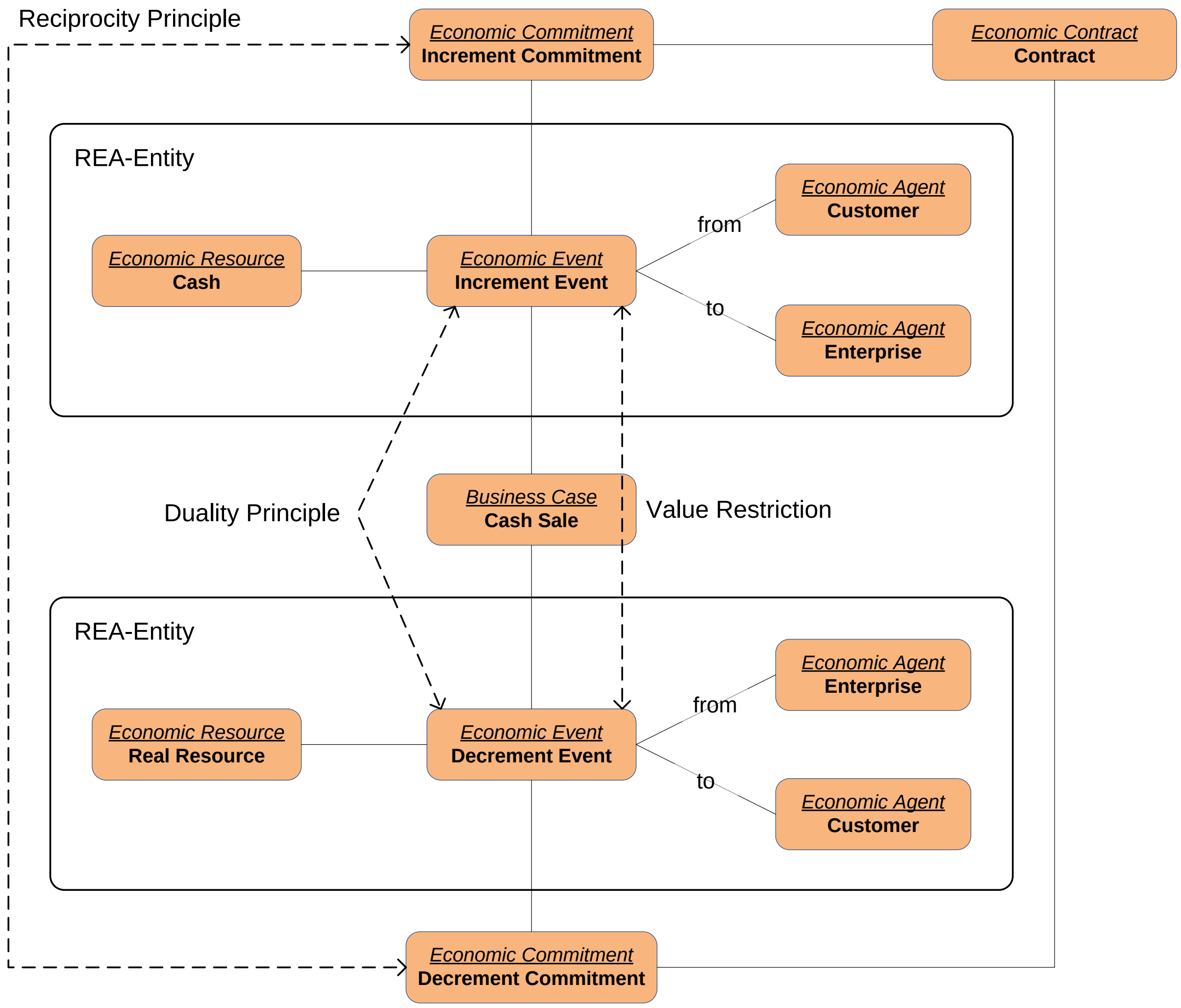


Figure 6: The REAv2 model including Events and Commitments

Figure 6 illustrates an extended REAv2 model. The increment and the decrement economic commitments can be fulfilled by economic events, but they need not to be fulfilled. The economic contract merges the commitments and collects the according agreements.

2.2. International Policies

Standards and rules are necessary to have the same base. Nowadays enterprises over the whole world have a heavy networking and their used IT systems have to work together. If every IT system would have its own database schema, techniques and processes, the integration of different systems would not be possible. Only with big effort of human and financial resources the systems could be linked up to exchange their data.

International organisations try to construct a base, which is the starting point for further developments of IT systems. In particular in the field of ERP systems, there are a few standards, which shall be considered when designing and developing a next generation ERP system.

2.2.1. AEO Standard

The accounting and economic ontology (AEO) standard with its designation ISO/IEC 15944 specifies the REA framework of McCarthy with its important concepts and elementary definitions in accounting and economics.

"This is a question of ontology – a formal specification of the concepts that exist in some domain of interest and the relationships that hold among them."[3 p. 5] Developments built on the REA framework do not only enable the collaboration of different IT systems, it is necessary for doing the next step to an automated data exchange.

The essential object as defined in the AEO standard can be identified by answering the following four questions:

- *Who is involved in the collaboration (Persons)?*
- *What is being exchanged in the collaboration (Economic Resources)?*
- *When (and under what trading conditions) do the components of the exchange occur (Economic Events)?*
- *Why are the trading partners engaged in the collaboration (duality relationships between resource flows)?*[3 p. 25]

Person

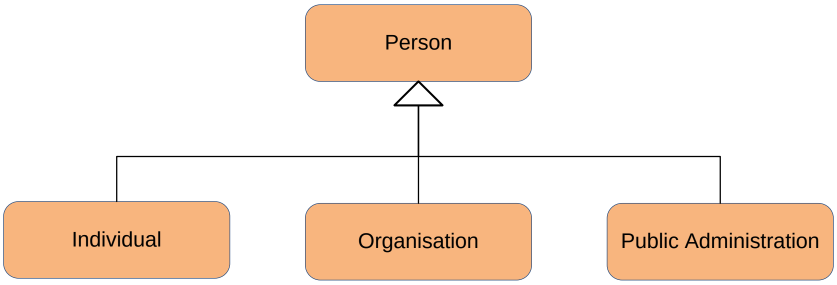


Figure 7: The person and its subtypes

A person can be an individual, an organisation or some kind of public administration, who takes part in a business collaboration. It is the requirement that the person is the owner and has the legal right of the resource which shall be exchanged in a business transaction. The generalisation of a person can be seen in Figure 7.

Economic Resource

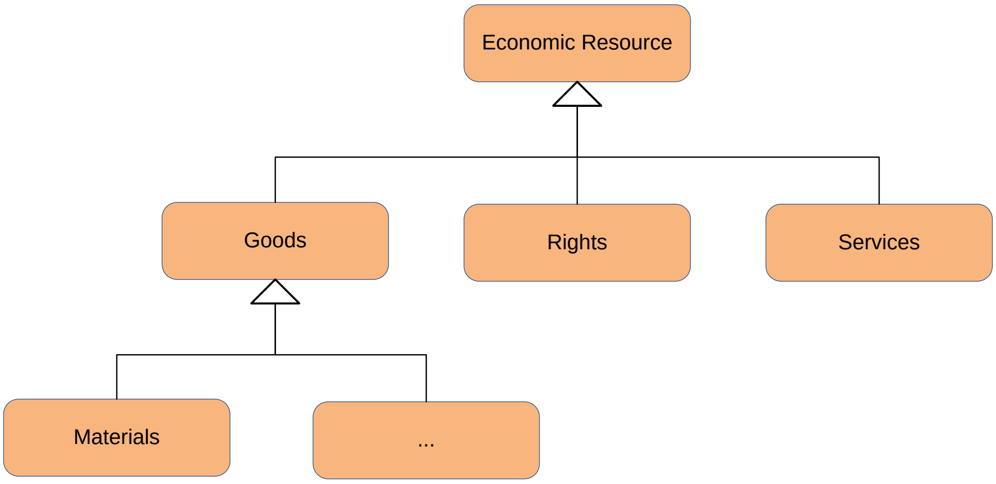


Figure 8: Economic Resources and its subtypes

An economic resource is scarce and under the economic control of a person for whom it has an economic value. Economic resources are exchanged between persons in business transactions. Figure 8 depicts some subtypes of the economic resource, which can be extended with individual classes, according to the requirements of the IT system to develop.

Economic Event

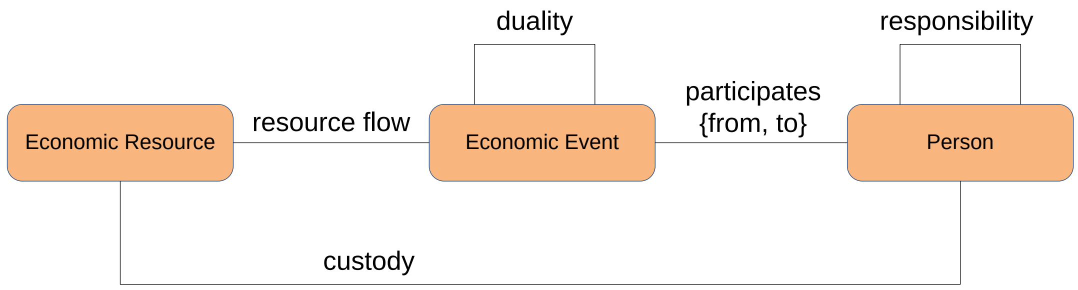


Figure 9: REA Entity according to the AEO standard

Figure 9 shows the REA entity according to the AEO standard. There are a quite few differences to the REA entity illustrated in Figure 4, but some have to be explained. The AEO standard defines a person in a more specific way compared to the economic agent. First, a person is responsible to other persons in the meaning of a hierarchical structure. The trading partners in a collaboration model shall know about the responsibilities in the enterprise, which is indicated by the self reflecting association. Second a person has a custody association to the economic resource which shall express the ownership between the person and the resource. A person can only sell resources which are held by the person. The other associations resource flow, duality and participates are already explained in Chapter 2.1.

Duality principle and Reciprocity principle

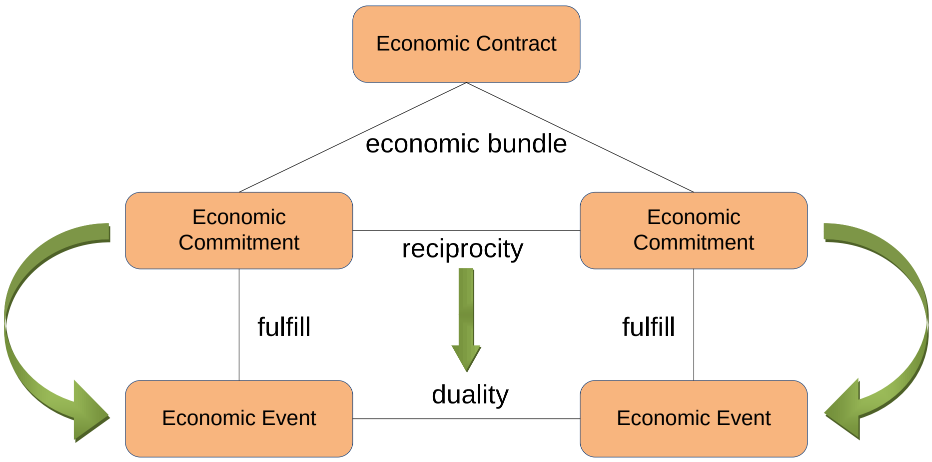


Figure 10: The Relationship between Reciprocity and Duality

The economic rules duality principle and reciprocity principle have the same task, but between different objects as depicted in Figure 10. The duality principle is the association between two economic events, where the one is the opponent of the other one in an exchange. The reciprocity principle is defined in equal measure but between economic commitments. So if a partner wants to execute a commitment, the other partner on the opposite has to do the same. Both commitments will be fulfilled and the reciprocity principle becomes a duality principle. The economic contract collects all concerning commitments and agreements for later usage.

2.2.2. ECSI Standard

The Enterprise-control system integration (ECSI) standard is an international standard about the collaboration of IT systems for enterprise management and for manufacturing execution systems. It has been worked out by the International Electrotechnical Commission (IEC)[3] on basis of the ANSI/ISA 95 standard, which has been developed by the International Society of Automation (ISA)[4].

Currently the ECSI standard with its designation IEC 62264 consists of three parts:

- Part 1: Models and terminology
- Part 2: Object model attributes
- Part 3: Activity models of manufacturing operations management

The intension of this standard is to have a basis for interoperability among the different IT systems in an enterprise. This is necessary because the enterprise management level uses other software systems than the manufacturing execution level. Of course a manager needs other software tools to make his decisions than IT systems need to control a manufacturing line.

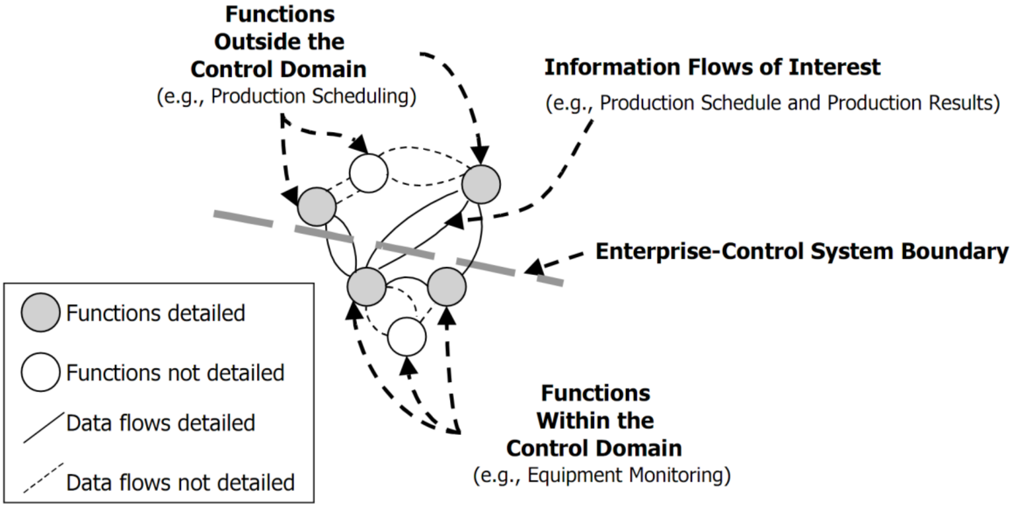


Figure 11: Enterprise-control system interface[7]

The information flow which is described in the ECSI standard, part 1 describes the interface between the functions within the control domain and those outside the control domain. Those

---

[3] http://www.iec.ch/

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

functions and information flows that are only outside or inside the control domain are not subject of the standard. In particular the information flows over the boundary and the functions which exchange information over the boundary are considered which is illustrated in Figure 11. All other data flows and functions are subject of other standards within a specific domain.

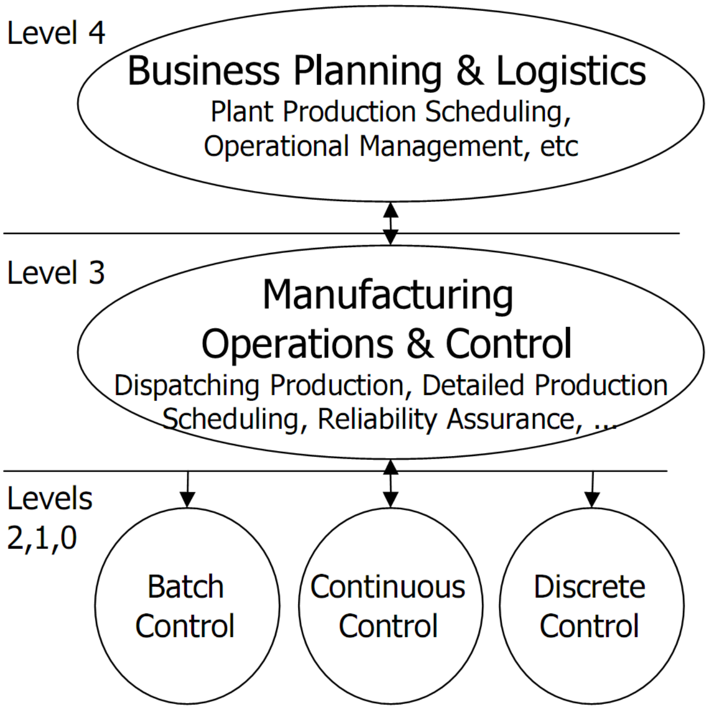


Figure 12: Functional hierarchy model[7]

To get more insight the functional hierarchy model can be seen in Figure 12. Five different levels are mentioned which can be found in the organisational structure of an enterprise. In each level different decisions have to be made. Part 1 and 2 of the ECSI standard define the interface with detailed information definitions between the business planning and logistics level and the manufacturing operations and control level. For example the production schedule for a period is created in level 4 and in level 3 this plan is broken down into single weeks. To exchange these data, the IT systems in both levels have to handle the same information, but with a different intention and they have to have standardised interfaces to send and receive the information objects, e.g. the production schedule. Part 3 of the ECSI standard focuses on level 3 and provides detailed activity definitions to handle the information flow within this level.

Figure 13 shows the functional model in which the boundary of the enterprise control interface is marked by the dotted line. The line can be compared to the level 3 / level 4 interface in Figure 12. The essential production functions can be found within the dotted circle whereas the functions outside belong to the control domain. The named associations are the important information flows between the functional units.

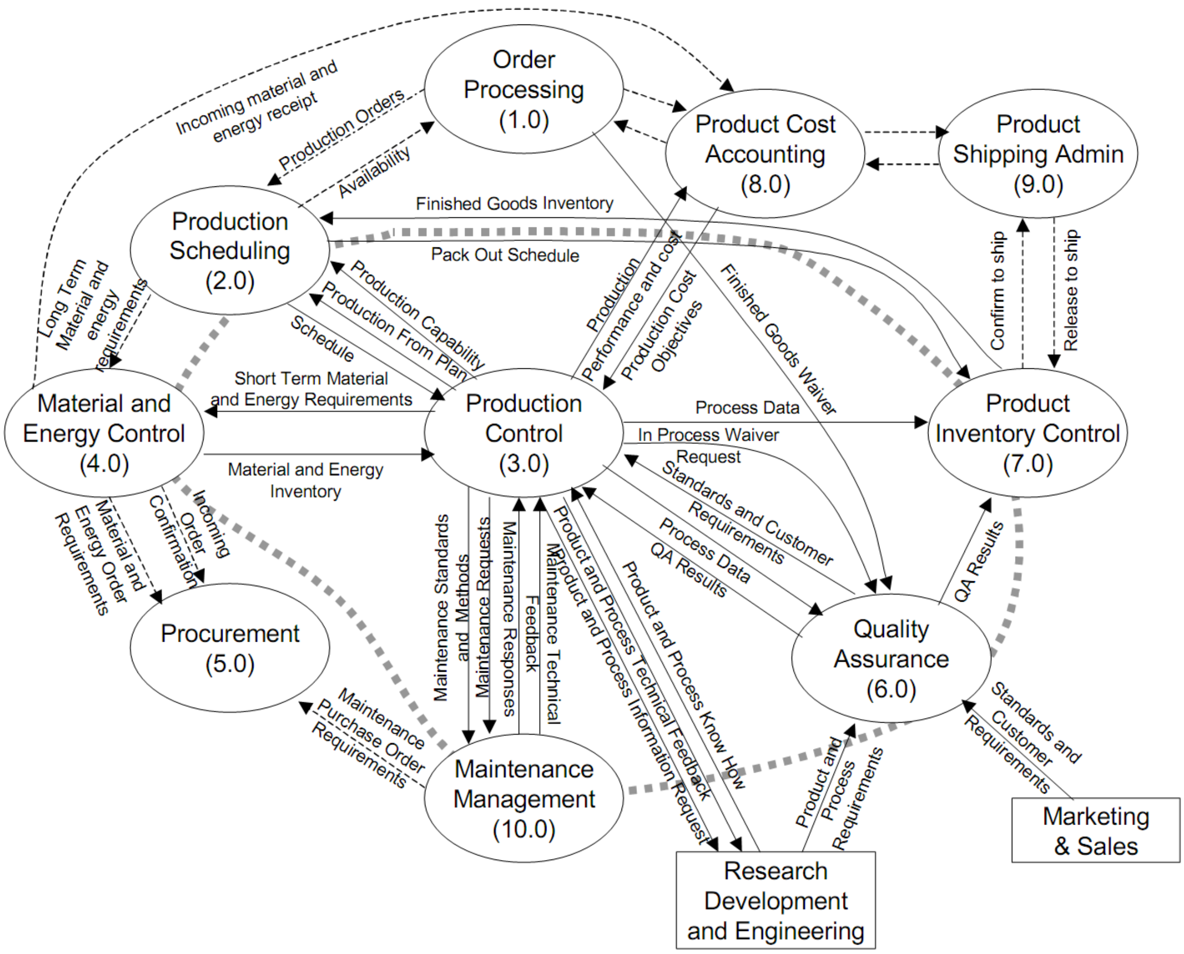


Figure 13: The Functional enterprise-control model [7]

The ECSI standard describes in detail which classes shall be implemented and how they are associated among themselves. Figure 14 illustrates the personnel resource model with the according classes. The class Person represents a single individual and can be categorised by the class Personnel Class, which describes a group of persons with similar attributes. The property classes can be described as a generic property concept to enhance the attributes of the associated classes Person and Personnel Class. The qualification test classes can be used to store information about particular specifications or about the results of the individuals.

The models in the standard describe the minimum requirements for the database schema, additional attributes, classes and associations can be used to extend the models, but with this standardised base the interoperability between the different IT systems is ensured.

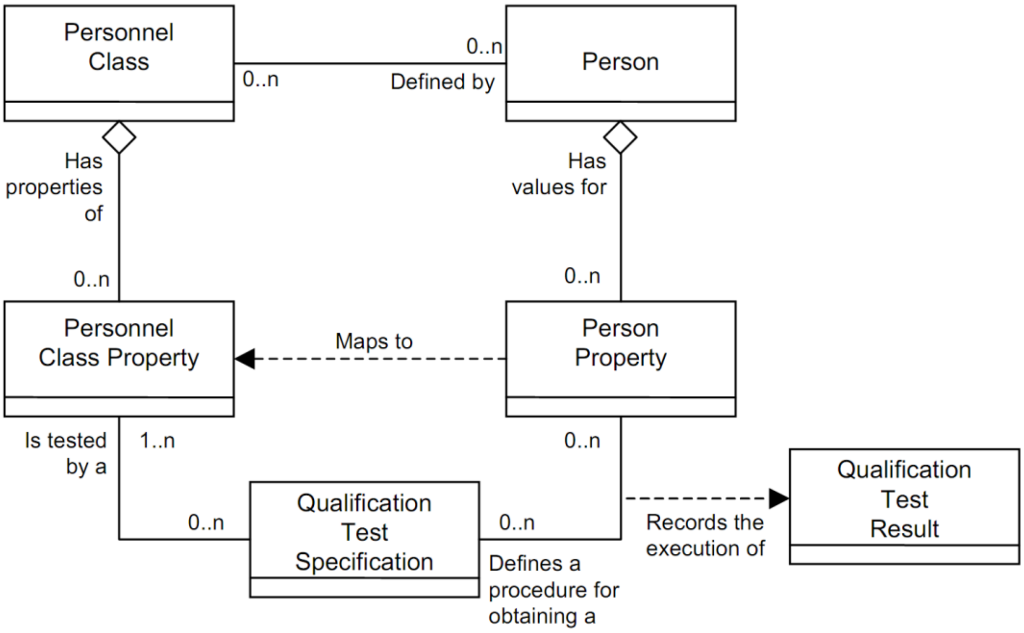


Figure 14: The Personnel resource model from the ECSI standard[7]

2.2.3. COSO II Framework

ERM deals with opportunities and risks which both may affect the business objectives, whereby an opportunity describes an event with a positive impact contrary to a risk which can have a positive or negative impact. For example the reliability of a device is a risk with a pure negative impact, because it will be determined if the device fails. The second type is a risk with a possibly positive or negative impact, e.g. the increase or decrease of the interest rate of an underlying financial resource. Both just mentioned risk types are expatiated and exemplarily implemented in Chapter 3.

To establish a law compliant, widely accepted and high usable guidance on internal control and against fraudulent financial reporting the Committee of Sponsoring Organizations of the Treadway Commission (COSO) was founded in 1985. COSO is an American organisation and its mission statement is defined as follows: *"COSO's mission is to provide thought leadership through the development of comprehensive frameworks and guidance on enterprise risk management, internal control and fraud deterrence designed to improve organizational performance and governance and to reduce the extent of fraud in organizations."* [8]

About a decade ago and after some big business scandals COSO observed the need of policies and rules for an organised risk management in organisations and entities. They started to develop an integrated framework for internal control, which assists managers to establish a robust risk management in their enterprises. The resulting framework provides important aspects as approved concepts, a unique language, key principles, etc. to improve the internal control in companies and guide them to a more comprehensive and integrated risk management.

Risk management is a process which shall help to identify, assess and manage various risks in an enterprise. Economic activity always deals with risk and uncertainty with the possibility to increase or decrease value, so COSO developed an integrated framework named COSO II which shall help to implement ERM in an enterprise and is defined as follows: *"Enterprise risk management is a process, effected by an entity's board of directors, management and other personnel, applied in strategy setting and across the enterprise, designed to identify potential events that may affect the entity, and manage risk to be within its risk appetite, to provide reasonable assurance regarding the achievement of entity objectives."* [9 p. 2]

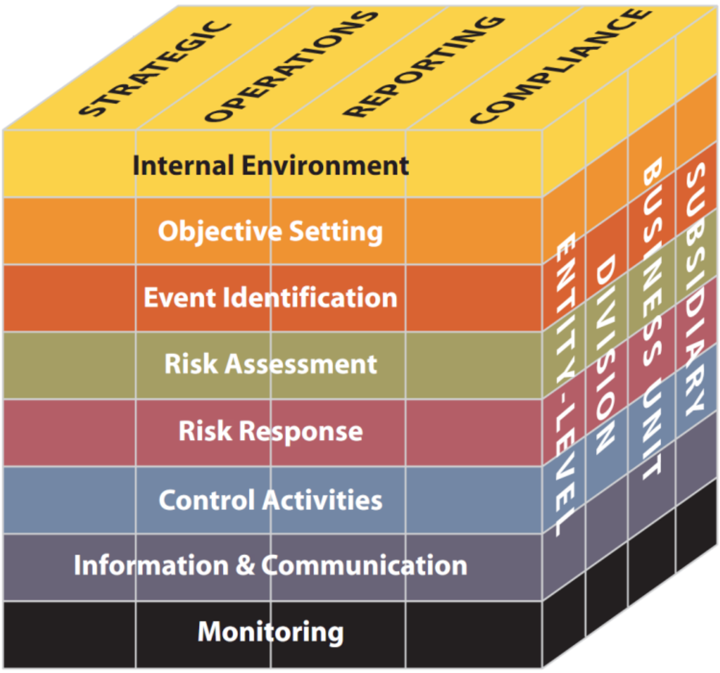


Figure 15: The compact illustration of COSO II

As one among many results of the COSO II development a cube was created, which is presented in Figure 15. This cube illustrates the relationship between objectives, ERM components and the entity´s units. The graphic shows the objectives placed on top of the cube, which are the business goals and shall be reached. The ERM components are those tasks in the horizontal rows that are needed to achieve the objectives. The inclusion of the entity´s units, which are shown in the third dimension, shall illustrate that ERM has to be implemented at every level in an organisation.

Because the focus of the ERM lies on the processes and their events the eight ERM components are the essential tasks in a risk management process. After the internal environment and the objectives have been set the risk measurement has to be executed, which assesses the events and their various risks. The results enable a forward-looking risk controlling with which possible future risks can be avoided and counteractive measures can be initiated in time. A more details explanation of feed-forward management processes is given in the following Chapter.

2.3. Management Models

ERM is a process, so a management framework is needed to support the circular flow of the risk management tasks. Plan-do-check-act (PDCA) is such a circular process, which intends to improve the management system continuously. Originally it comes from the quality management and was early mentioned as “Shewhart cycle”.

The PDCA framework aims to establish an improving management system, which is done by repetitive cycle runs and continuous adjustments during the iterations. When solving a problem, leading a project or managing a risk, the four PDCA phases are the same:

- Plan – the process has to be planned. Planning means to identify the events, defining the goals and creating an action plan. Every planned step shall also be documented for later analysis.
- Do – the execution of the plan. In particular the execution of the plan is tried, because no plan is perfect and so some problems may occur. In such a case the problems and the way, how the problems were responded, have to be recorded.
- Check – analyse and improve the process. After the Do-phase the good and the poor results have to be checked in order to improve the process. If there is some deviation of the focused goal, the cause has to be determined. This avoids similar problems in the next run. As a result the process will become more efficient and the achievement of the goals is getting closer.
- Act – adjust the process. The root cause of the identified problems has to be eliminated by correcting the current tasks. If a deviation was determined, the process has to be adjusted to get back to the right way. If all change requests are applied the improvements of the process has to be communicated to all stakeholders.

The PDCA framework can be implemented as a reactive or a proactive management system. The reactive management system is a feedback process and compares realised and planned values. In contrary the proactive, feedforward process compares predicted with planned values.

A further distinction of PDCA frameworks is the type of implementation, which can be a closed or an open loop management system. The difference between both is the check phase, which is missing in the open loop management as depicted in Figure 16. ERM is a closed loop management system, because the result of the process is compared to given limits and if there is some deviation, the adjustment will be part of the following iteration.

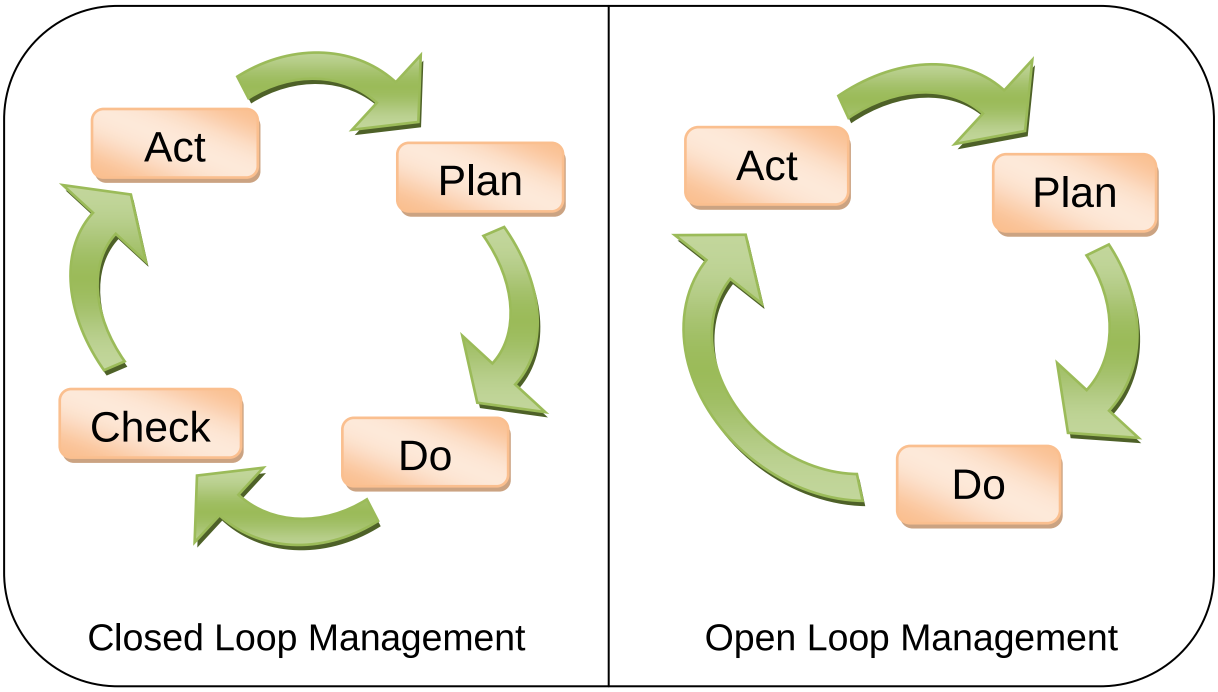


Figure 16: PDCA framework as closed and open loop management system

2.4. ERPControl – a REA-MIS Implementation

ERPControl is a prototypical implementation of a REA-based management information system (MIS). The target of the development can be shortly described by the concept "IT follows Business". The conceptualisation, development and implementation are put into practice by the Department of Corporate Finance, Planning and Control as part of the Institute of Management Sciences at the Vienna University of Technology.

2.4.1. Technology Components

Software Architecture

ERPControl integrates established industrial standards and is a web 2.0 application, which is built upon the Java Platform, Enterprise Edition (Java EE), Hibernate and the JBoss Seam framework (Seam). The key features for choosing these frameworks were:

- Business logic through Session Beans (SB)
- Persistence through Entity Beans (EB)
- Object-relational mapping (ORM) through Hibernate
- Process orientation through the JBoss Business Process Management (jBPM)
- User interface through JavaServer Faces (JSF) and JBoss RichFaces (RichFaces)
- Integrated development environment (IDE) through the Eclipse platform

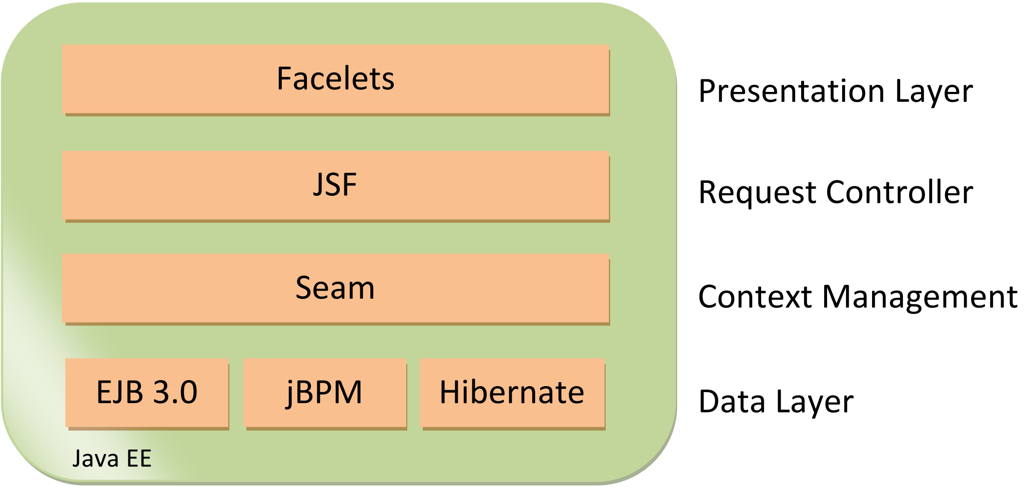


Figure 17: Java EE supports several other frameworks for ERPControl

Figure 17 depicts the composition of several frameworks provides the necessary functionality for implementing the theoretical concepts into an ERP system. The persistence into a physical database is done by Hibernate, which is an ORM framework for Java. The usage of an ORM framework reduces the programming effort of the simple create, read, update and delete (CRUD) operations in the database. The data objects are implemented by EB, which are part of the Entity JavaBeans 3 specification (EJB3). Each EB represents a single data record and each EB class defines a special table in the database.

The jBPM can also be found in the data layer, because of its long-running business processes, which can even last up to a year. To hold their process state for such a long period, they are persisted to the database and restored as required, so actually a server shutdown cannot interrupt them.

Seam provides the context management for the ERP system, so different components, contexts and concepts are provided through this framework. A more detailed description will follow in this Chapter. The JSF framework manages the user interaction, offers some valuable techniques for data handling and supports input validation. Facelets is an open source template framework, which converts the processed extensible mark-up language (XML) documents into an user interface.

The distinction into data, logic and presentation follows the programming paradigm separation of concerns. This special design pattern is named Model-View-Controller architecture (MVC) and assists developers by isolating the business logic from the user interface and from the underlying database.

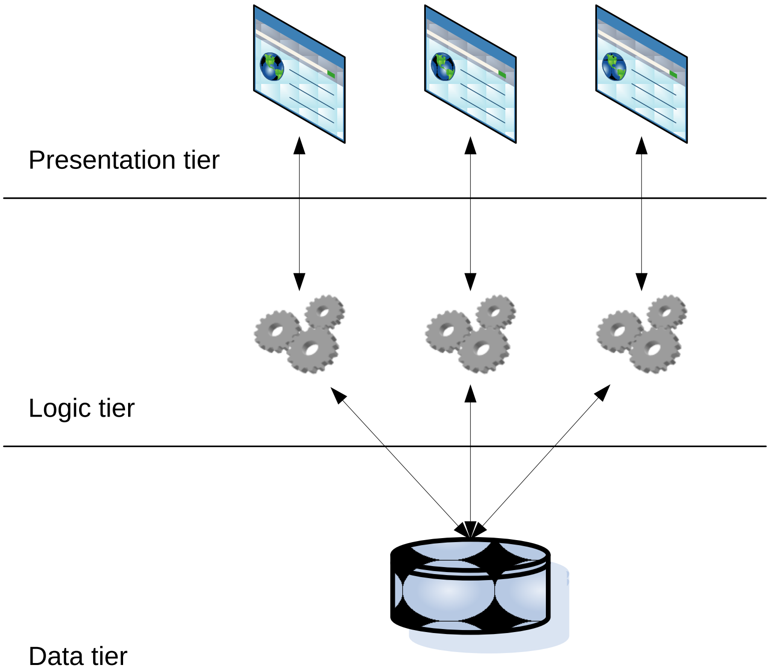


Figure 18: ERPControl is a tree-tier web-application

For example the exchange of methods, the reengineering of the frontend or the replacement of the physical database does not affect the other parts of the software system. Figure 18 illustrates the connection between the three tiers, whereas the objects are fetched from the database (data tier) to process the required information (logic tier) which then are presented at the front end (presentation tier).

Implementing the Logic – Session Beans

Each framework provides a specific functionality, which can be used in an easy and quick way by annotating the according variables, methods or classes. A detailed manual can be found in [10]. Annotations can be compared to XML: the semantic is provided by the XML tags, which mark up the pure text. Analogous annotations extend their reference objects to powerful components with special functionality and automatism in these frameworks. In the following code fragments the most frequently used annotations are presented, starting with the SBs.

Code 1: Class annotations

```
package at.ac.vut.bpmis.jbpmorchestration;
@Stateful
@Name("riskEventManager")
public class RiskEventManager implements IRiskEventManager { ... }
```

A class annotation extends a plain, old Java object (POJO) that it becomes a special component in the system. The SB are located in different packages of ERPControl, e.g. the

SFSB `RiskEventManager` is part of the `jbpmorchestration` package. A short description of the annotations from Code 1:

EJB3 annotation: `@Stateful`

This annotation marks a POJO to be a SFSB. The public methods can be invoked by other SB or in the graphical user interface (GUI).

Seam annotation: `@Name("riskEventManager")`

The annotated class can be invoked by other components via the name in the brackets.

A second technique can be seen in Code 1: the SB implements a public interface. This interface defines the methods, which can be invoked and which must be implemented by the SB.

Code 2: Variables annotations

```
/** The interface of the SLSB for the risk evaluation. */
@EJB
private RiskEvaluation riskEvaluation;

/** The list for showing the RiskEventCategory-s. */
@DataModel
private List<RiskEventCategory> riskEventCategoryList;

/** The selected RiskEventCategory from the according list. */
@DataModelSelection(value="riskEventCategoryList")
private RiskEventCategory riskEventCategorySelected;

/** The RiskEventCategory ID for the bijection in the BP scope. */
@Out(scope=ScopeType.BUSINESS_PROCESS, required=false)
@In(scope=ScopeType.BUSINESS_PROCESS, required=false)
private Long riskEventCategoryId;

/** The starting Date for the RiskEvent reliability. */
@Past
private Date startDateTemp;
```

Several different annotations for class variables are shown in Code 2:

EJB3 annotation: `@EJB`

Through this annotation another SB is injected in the current class. In the above code fragment the SB `RiskEvaluation` is injected. That allows the usage of already implemented business logic for the calculation.

Seam annotation: `@DataModel("name")`

The data model can be some type of collection, which is outjected for presentation in the GUI. In Code 2 the collection is a list of `RiskEventCategory` objects and can be invoked by JSF through the specified name. If no name is denoted, the name of the variable is the component´s name.

Seam annotation: `@DataModelSelection(value="riskEventCategoryList")`

The data model selection marks the variable, which will get a selected element injected. The value specifies the according data model, in which the user can chose an element of a collection.

Seam annotation: `@Out(scope=ScopeType.BUSINESS_PROCESS, required=false)`

Seam annotation: `@In(scope=ScopeType.BUSINESS_PROCESS, required=false)`

These annotations specify that the variable should be injected respectively outjected in the according context. Outjecting means that the component can be invoked by other parts of the system, e.g. by JSF in the GUI. Injecting means therefore that the variable in the class is set by the according component, which is taken from the context. Both annotations are called bijection, which is an innovation from Seam and empowers the developer to build more complex systems in less time. The attribute scope specifies which context should be taken for the bejection. The attribute `required` instructs the system, if the variable is optional or not.

Hibernate annotation: `@Past`

A variable with the type `Date` can be marked with this annotation, which enables an immediate validation of the user input. The input value for this annotated variable must be a date, which has to be in the past.

Code 3: Method annotation for the data model

```
/** Factory method for displaying the RiskEventCategory-s. */
@Factory("riskEventCategoryList")
public void displayRiskEventCategories() {
    riskEventCategoryList = entityManager.createQuery("from
RiskEventCategory rec where rec.closed = false").getResultList();
}
```

Code 3 shows a factory method:

Seam annotation: `@Factory("riskEventCategoryList")`

The factory annotation marks a method which fills an empty data model. When a data model is invoked for the first time, it will not contain any elements. The system looks for a factory method, which has the name like the data model component. If such a method is found, the collection will be filled with the provided information from the factory.

Code 4: Method annotations for business process management

```
/** Method for initializing a new BusinessProcess. */
@CreateProcess(definition="bp_riskEventAdd")
public void initialiseAdd() {
    ...
}

/** Starting method for a BusinessProcess Task-Node. */
@StartTask
public String beginCategorySelection() {
    ...
}

/** Finishing method for a BusinessProcess Task-Node. */
@EndTask(transition="next")
public void endCategorySelection() {
    ...
}
```

In Code 4, the annotations to organise the business processes are presented:

Seam annotation: `@CreateProcess(definition="bp_riskEventAdd")`

When this annotated method is invoked, the process with the specified name will be initialised by the jBPM.

Seam annotation: `@StartTask`

The execution of this method starts the according task-node, which is the next in the actual business process.

Seam annotation: `@EndTask(transition="next")`

This annotation instructs the system to finish the current task-node in the running process and to forward to the next task-node as denoted by the attribute transition.

Code 5: Method annotations for page flow management

```
/** Starting method for the according PageFlow. */
@Begin(pageflow="pf_riskEventEdit")
public String beginnEditRiskEvent(Long reId) { ... }

/** Finishing method for a PageFlow. */
@End
public void endEditRiskEvent() { ... }
```

Analogous to the business processes, the page flows can be directed by annotations as illustrated in Code 5:

Seam annotation: `@Begin(pageflow="pf_riskEventEdit")`

The `begin` annotation instructs Seam to start a new page flow. A page flow is an atomic sequence of steps, which in the sense of an atom are not separable. These sequence must be stepped through till the specified end or it can be aborted as denoted in the page flow. The name of the page flow is quoted in the annotation attribute.

Seam annotation: `@End`

A method with this annotation instructs Seam to finish the current page flow. The next page can be specified in the return value of the method.

Implementing the Persistence – Entity Beans

The SBs include the logic for the dynamic in the system, the methods process data to create new information or they manipulate the EB to record the changes in the system. The EBs provide the structure, which defines how the information is persisted in the database. The database schema is generated out of the class structure of the EBs, the data manipulation is done by CRUD methods, which is also a essential part of the EB. The Hibernate framework is responsible for the ORM and organises the persistence management with a number of annotations [11]. Here are some examples of annotations in EB:

Code 6: Class annotation

```
package at.ac.vut.bpmis.bo.erm;
@Entity
public class RiskEvent {

    /** The id. */
    private Long id;

    /** The description. */
    private String description;

    ...
}
```

An EB is created by annotating a simple POJO:

EJB3 annotation: `@Entity("name")`

This annotation specifies that the class is an EB and therefore persisted in the database. The annotation attribute is optional, if there is no name annotated as shown in Code 6, the class name is taken for the name of the EB.

The variables of an EB are by default the columns of a database table. Of course the mapping can be done with a lot of additionally annotations, but for normal, the default values are sufficient for generating a suitable database schema.

Code 7: Association annotations

```
/** The RiskEventCategory-s, to which this RiskEvent belongs. */
private Set<RiskEventCategory> categories =
     new HashSet<RiskEventCategory>();

/** Setter method. */
@ManyToMany
public Set<RiskEventCategory> getCategories() {
     return categories;
}

/** Getter method. */
private void setCategories(Set<RiskEventCategory> categories) {
     this.categories = categories;
}
```

In Code 7 the necessary definitions for associating the EB are illustrated. The variable `categories` connects the EB `RiskEvent` with its `RiskEventCategory`s as defined in the data model. To specify the association type, the setter method gets an annotation:

EJB3 annotation: `@ManyToMany`

This annotation indicates that many `RiskEvent`s can be associated with many `RiskEventCategory`s and vice versa. In the database, an additional table will be created to provide this bidirectional connection. Available association types are furthermore `OneToOne` and `OneToMany`.

Implementing the Dynamic – Business Processes

SBs include the logic for the dynamic, so the dynamic itself has to be orchestrated beside the business logic. From a technical view a jBPM business process specifies some important details about this orchestration [12]. First, which tasks have to be executed in which order. Second it checks the authorisation of the participations, so several roles can be defined and each role has special user rights. Third the according framework provides a wide spectrum of possibilities for collaboration within a unique enterprise and outside to other partners.

Figure 19 depicts a simple business process, the name of the steps are according to the syntax of the jBPM. Starting point is a start-state, which brings the process directly to the first task-node. During the life-time of a business process, typically several task-nodes have to be completed, whereas special nodes like decisions and forks can extend the possibilities to bring the process to the end. Finally, if the end-state is reached, the business process will be closed after all finishing tasks are completed.

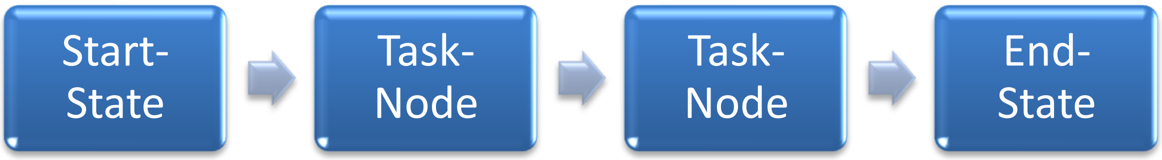


Figure 19: A simple business process example

The tasks of the jBPM handle the different instances of the business processes, orchestrate between the succeeding task-nodes and persist the states of each business process to the database. jBPM business processes are called long-living processes, because their state can even be reconstructed after a server reboot.

Code 8: jBPM business process definition in XML

```
<?xml version="1.0" encoding="UTF-8"?>
<process-definition name="bp_NameOfTheProcess">

    <start-state name="start">
        <transition to="first_TaskNode"></transition>
    </start-state>

    <task-node name="first_TaskNode">
        <task name="taskNode_Name">
            <description>
                a textual explanation what is done
            </description>
            <assignment actor-id="#{actor.id}"/>
        </task>
        <transition to="second_TaskNode" name="next"></transition>
        <transition to="end" name="cancel"></transition>
    </task-node>

    ...

    <end-state name="end"></end-state>

</process-definition>
```

The extract of Code 8 shows the beginning of an XML defined jBPM business process. The process can be instantiated by the jBPM through invoking its `name` as denoted in the `process-definition`. After the process initialisation the `start-state` is searched, which defines the `transition` to the first `task-node`. The process instance waits in the `task-node "first_TaskNode"` for user interaction. The starting and ending of a task-node is done through the SB annotations as explained above. The `@End` annotation specifies the transition, which has to be taken next. In the above SB example the method `endCategorySelection()` is annotated with `@EndTask(transition="next")`, therefore process instance switches to the `task-node "second_TaskNode"`, where it waits again for user interaction. At the end the `end-state` instructs the jBPM that the end of the process is reached and to close the business process instance.

Interaction with the User – Graphical User Interface

ERPControl is a Web 2.0 application, which means that the user can interact with the system via a normal web browser. The JBoss application server publishes the webpages, manages the business processes, starts and ends the pageflows and invokes the session beans. A detailed explanation of these techniques will follow in Chapter 4.

Figure 20 presents a screenshot of the start page of ERPControl. The page is structured in three parts: at the top of the page a status bar is displayed, where the logged-in user can be seen. At the left side the menu is positioned, the menu elements can be expanded to see the sub-elements. The main part of the screen is used for presenting the content.

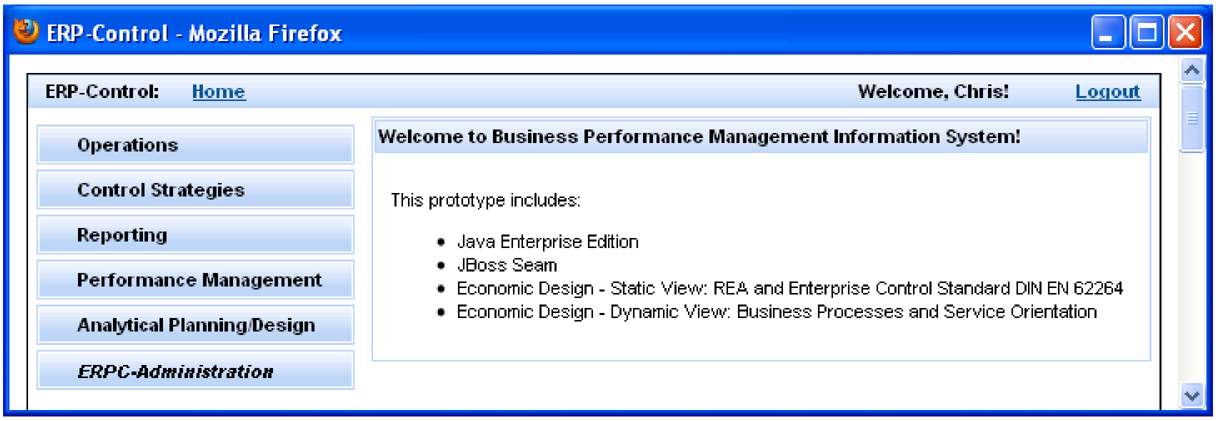


Figure 20: ERPControl – a web 2.0 application

The JSF framework supports the development of the user interface and is part of Java EE. According to the presented MVC architecture in Figure 18 JSF enables the developer to focus on the application logic by assembling the webpages from components of the logic tier. This speeds up the implementation of the presentation tier and reduces therefore the problematic interfaces, which have a high fault potential.

RichFaces is an extension of JSF and uses templates to build the structure of the GUI. When a webpage is constructed the according areas are replaced by the actual content. The content-independent areas like the menu and the status bar have only to be defined once for all webpages. Only the content part has to be implemented, the other sections are inserted automatically during the generation of the webpage through the application server. The schematic compounding of the different areas of a webpage can be seen in Figure 21.

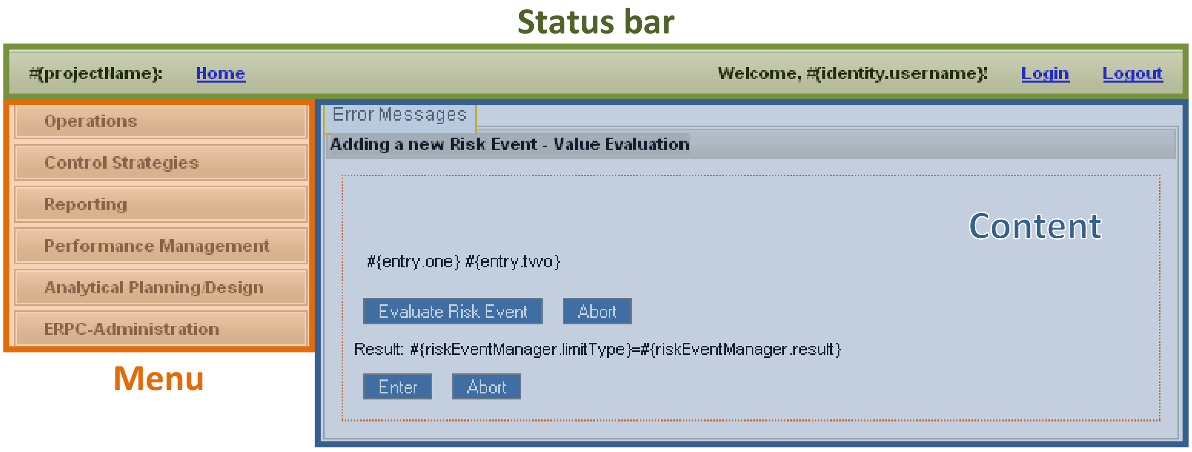


Figure 21: Template structure of the GUI

Furthermore RichFaces provides the functionality for up to date user interaction by the integration of Asynchronous JavaScript and XML (AJAX). So user inputs can be validated just after the user has entered the value. This speeds up the workflow and empowers ERPControl to have a similar behaviour like a desktop application.

Eclipse platform

The Eclipse framework[5] is a powerful software development tool including an IDE and a great number of plug-ins, which expand the possibilities to nearly all kind of software technology. It is written in Java and is used mostly to develop Java software projects, but other programming languages can also be implemented through the assistance of the plug-ins.

Figure 22 presents the Eclipse IDE during the ERPControl implementation. The application is separated in different frames and each of them provides a special task or aspect of the development. Code details can be displayed during the programming, the according pop-up info is shown in the middle of the screenshot.

Another big enhancement for the implementation is the integration of the JBoss application server (JBossAS) in the Eclipse framework. Via a plug-in the JBossAS can be started and ended by Eclipse. This speeds up the testing of new functions and procedures, because after each change of a SB or an EB, a jBPM business process, a webpage or just a little validation annotation the JBossAS has to be restarted to publish the new code on the webserver.

---

[5] http://www.eclipse.org/

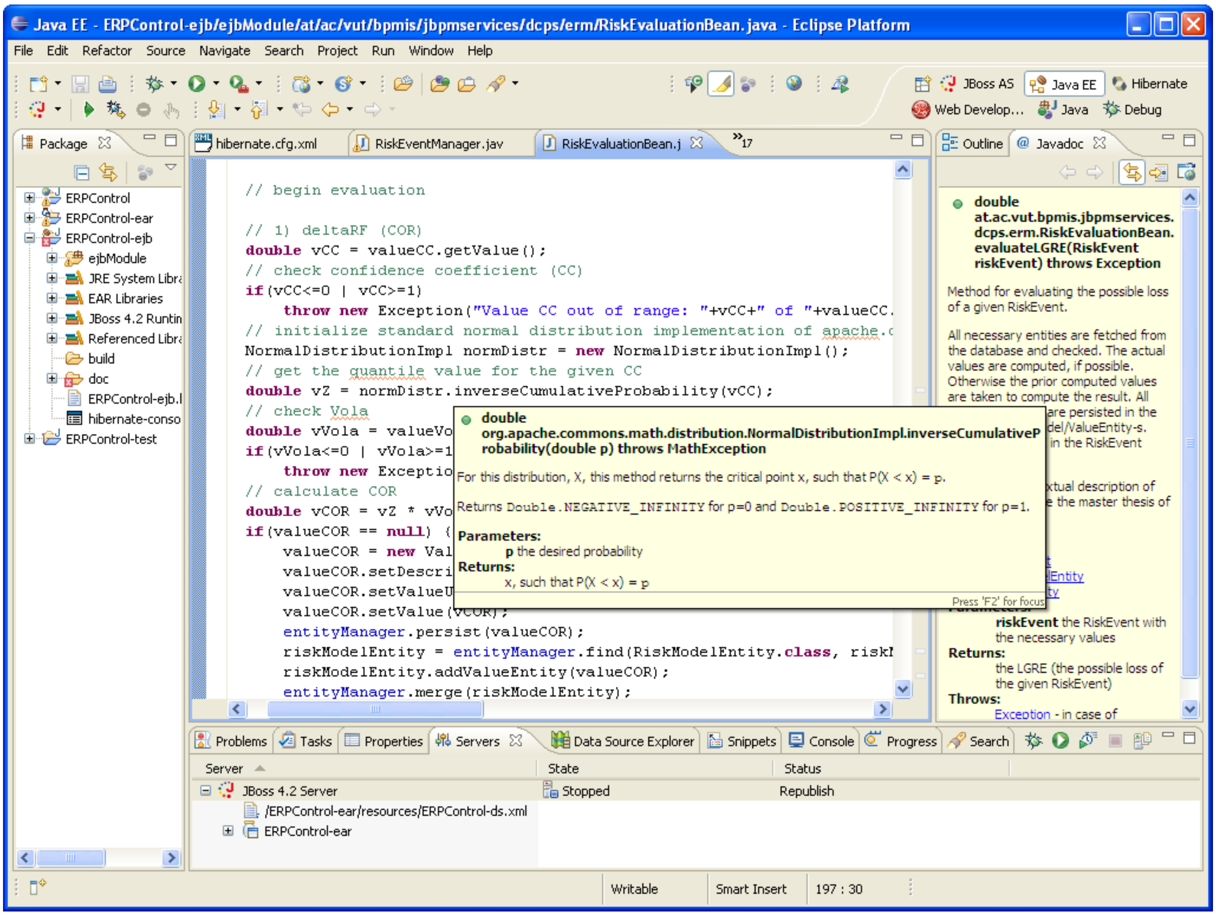


Figure 22: Eclipse IDE for the development and implementation of ERPControl

2.4.2. Economic Aspects

The connection between the technological and the economic view can be seen in Figure 23. The economic entities are presented like the technical layers in Figure 18. The business objects (BO) refer to the EB in the data tier, the business activities (BA) are implemented through the SB in the logic tier and the business processes (BP) stand for the jBPM respectively the JSF/RichFaces in the presentation tier. All layers together build the business model, which represents the economic core of ERPControl.

The BOs define the structure of the data, which is a big difference between ERPControl and other traditional ERP systems. The resources are the starting point for the creation of a balance sheet or an income statement. The BOs are manipulated by the logic of the BAs, which provide the economic concepts, policies, rules and execute them in form of implemented procedures. The BPs orchestrate the BAs and define who may invoke a certain BA at which process step. So the BPs ensure that the BOs are manipulated by the BAs according to the implemented legal guidelines.

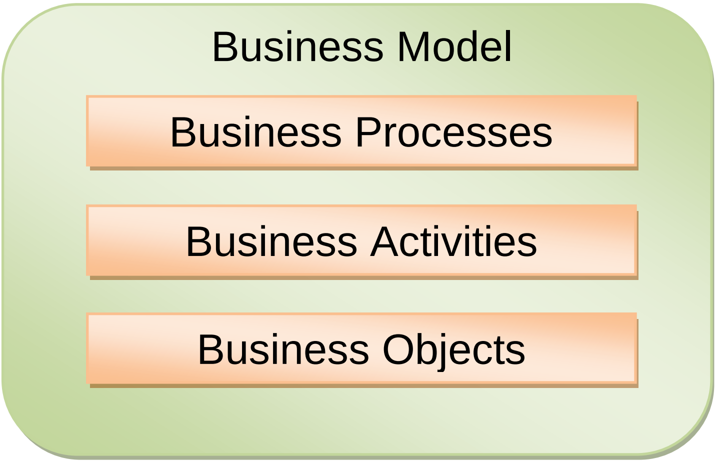


Figure 23: The business model in a technical view

Some packages of the ERPControl database model are shown in Figure 24. The `at.ac.vut.bpmis.bo.rea` package includes the core classes for the economic logic of ERPControl, namely the classes `EconomicResource`, `EconomicEvent` and `EconomicAgent`. These classes are fixed associated as described in the Figure, so an `EconomicEvent` is fixed linked with one `EconomicResource` and two `EconomicAgent`s, but with two separate associations (`fromAgent` and `toAgent`). Further the `EconomicEvent` must be part of a `BusinesCase`, represented by the composition. The `personnel`, the `goods` and the `financial` package extend the `EconomicResource` by the inheritance connection. For example the class `EconomicResource` is a generalisation of the subclass `MaterialDefinition`, which is located in the package `at.ac.vut.bpmis.bo.goods.material`.

An example of a business case implementation in ERPControl can be seen in Figure 25. The `EconomicAgent` customer buys a real resource from the `EconomicAgent` enterprise. From the view of the enterprise the `EconomicResource`, which is purchased by the customer, is decremented, because the property of the resource switches from the enterprise to the customer. On the other side the `EconomicResource` cash is incremented, because the enterprise gets the value of the sold resource back in form of cash. Both `EconomicEvent`s are associated with the `BusinessCase`, which is the central point of every economic activity.

Every economic exchange of resources is recorded by the BO `BusinessCase` and the `EconomicEvent`s. Thereby the `EconomicEvent`s represent the dynamic in the enterprise, so the income statement is compounded by the change in value of all `EconomicEvent`s for a defined period. The actual, available quantity of a resource can be queried from the BO `EconomicResource`. The quantity, the unit of measure, the price etc. are attributes of the resource.

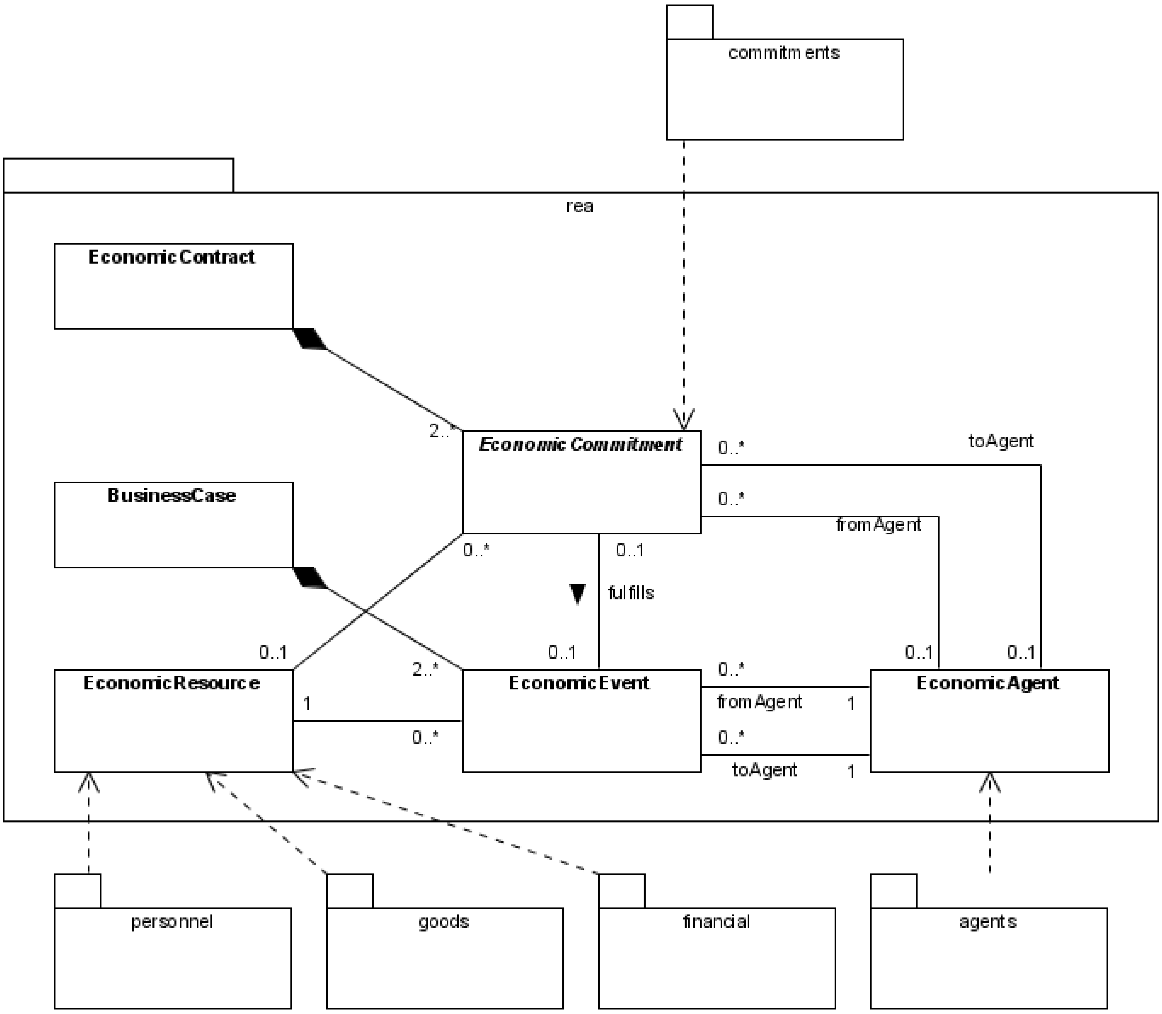


Figure 24: The rea package and the associated subpackages

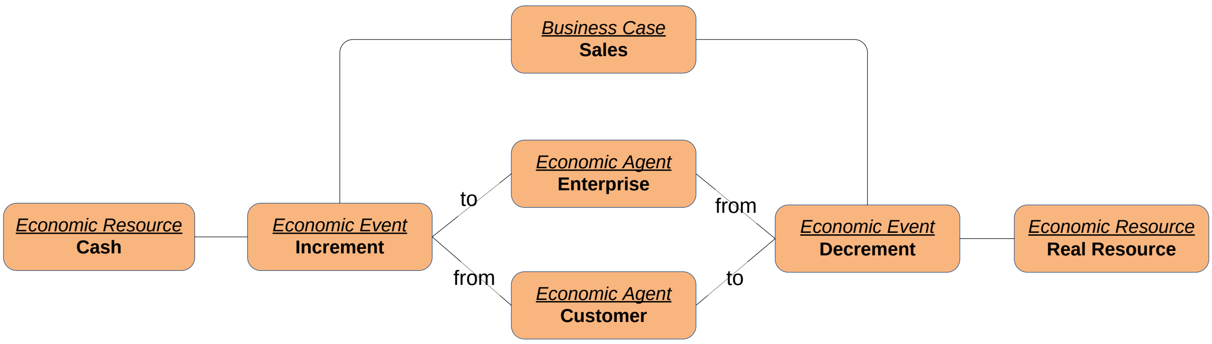


Figure 25: The business case Sales modelled as class diagram

3. Core Functionality for ERM Reports

When considering risk management in traditional ERP systems, it is mostly implemented by add-ons in the working ERP system or as an external tool, which has to extract the data expensive from the database to do the processing. The more consistent way is to integrate the ERM into the MIS right from the beginning.

ERM is not a single task, it has to be a full system-integrated and long-running process. Only then the user can use the whole spectrum of risk management in connection to the underlying resources to make his decisions.

Two concepts of an integrated dynamic enterprise risk management are presented at the following pages. The calculations to manage the risk are described in this Chapter, their implementation in ERPControl are subject of the Chapter 4. The first type of a dynamic ERM focuses on resources with a probability of failure and their change in time. The second type concentrates on fixed income financial instruments and the maximal loss in case of a change of the interest rate.

3.1. Dynamic ERM – Reliability model

When devices are devoted, the user wants to know, how long a special device is functional respectively how reliable it is. As described in [13] *"… reliability is the ability of an item (product, system, … etc.) to operate under designated operation conditions for a designated period of time or number of cycles."* Monitoring the reliability of resources enforces the processes of the enterprise risk management to notify the user, who can set actions before a failure occur. Certainly the value is the result of a stochastic process and therefore only some kind of recommendation but it helps to keep an overview of the reliability of the monitored resources supported by the tools and possibilities of the ERP system.

Every resource, which is used in some way in an enterprise, has a probability of failure. In case of a machine this could be a breakdown or a malfunction which leads to less quality of the output. If the machine is part of a production line, the manufacturing stops and a financial loss will occur. To avoid such a scenario the actual reliability of machines, vehicles, IT hardware or other similar resources can be estimated to maintain the machine or the replace the vehicle just before the damage happens.

The reliability is computed by the average life-time, which is taken to evaluate the reliability model over their whole life-time. As time goes by the actual reliability can be estimated by the risk model to arrange further actions. E.g. if the reliability decreases under a defined limit the user of the ERP system will be notified to induce some corrective measures.

3.1.1. Risk Model

The risk model, which is introduced to manage this type of risk, is widely used to model a random variable which represents a time-to-failure of the considered device. To create the risk model the exponential distribution is used. Based on the explanations in [13] the equation for the distribution function $F(t)$ is shown in (1).

$$F(t) = 1 - e^{-\lambda \cdot t} \qquad t \geq 0,\ \lambda > 0 \tag{1}$$

This distribution function for the probability of failure of a device is a function of time and expresses the probability that the considered device will fail sometime up to the time t. According to this, it is also called the unreliability of the device. To find the reliability function $R(t)$ instead of the failure distribution, the formula is converted as presented in (2).

$$\begin{aligned} R(t) &= 1 - F(t) \\ &= 1 - (1 - e^{-\lambda \cdot t}) \\ &= 1 - 1 + e^{-\lambda \cdot t} \\ &= e^{-\lambda \cdot t} \end{aligned} \tag{2}$$

In Figure 26 the graphical representation of the exponential distribution function in comparison to the reliability function is shown.

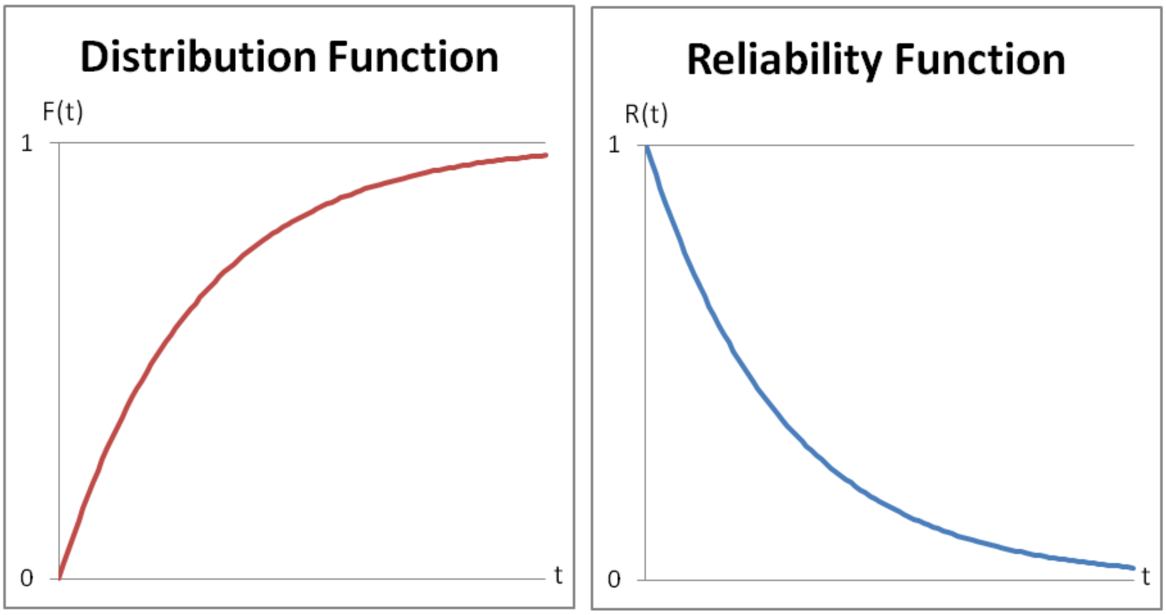


Figure 26: Graphical illustrations of the exponential distribution function and the reliability function

In the exponential distributed reliability model the failure distribution $F(t)$ has a single parameter, the failure rate λ, which is the frequency at which failures occur per unit of time. The failure rate can be evaluated by the reciprocal value of the average life-time (or expected value) $E[t]$ as described in (3). To get the failure rate, expert or sample estimators are used to determine the average life-time.

$$\lambda = \frac{1}{E[t]}$$

(3)

With the reliability function the present reliability of resources can be approximated. By default the parameter t in the reliability function $R(t)$ is set to one, because the risk horizon of one year shall be considered.

The risk horizon is the period for which the reliability is calculated and is fixed for the created risk event to compare the results of different dates. Certainly it is possible to create more risk events with different values of the risk horizon, e.g. a short-term risk event with the period of a quarter and a long-term risk event with the period of a year.

If the risk horizon is set to the average life-time, then the average life-time $E[t]$ is the characteristic life-time of all exponential distributions as explained in (4).

$$\begin{aligned} R(t) &= e^{-t \cdot \lambda} \qquad with: \; t = E[t] \\ &= e^{-E[t] \cdot \lambda} \\ &= e^{-E[t] \cdot \frac{1}{E[t]}} \\ &= e^{-1} \\ &\approx 0{,}37 \end{aligned}$$

(4)

This means that every resource reaches its average life-time with approximately 37% according to the exponential distributed reliability model.

3.1.2. Evaluation

A graphical representation of an example calculation of the reliability model can be seen in Figure 27. The example was calculated with an average life-time of six years, which is the green bar $R(E[t])$ in the diagram. The blue curve $R(t)$ is the reliability function with the failure rate $\lambda=1/6$ as described above. The purple bar $R(1)$ is the evaluated reliability with the default value for the risk horizon $t=1$. Additionally the *limit* of 50% is displayed in the diagram, which is mapped as the red dotted line.

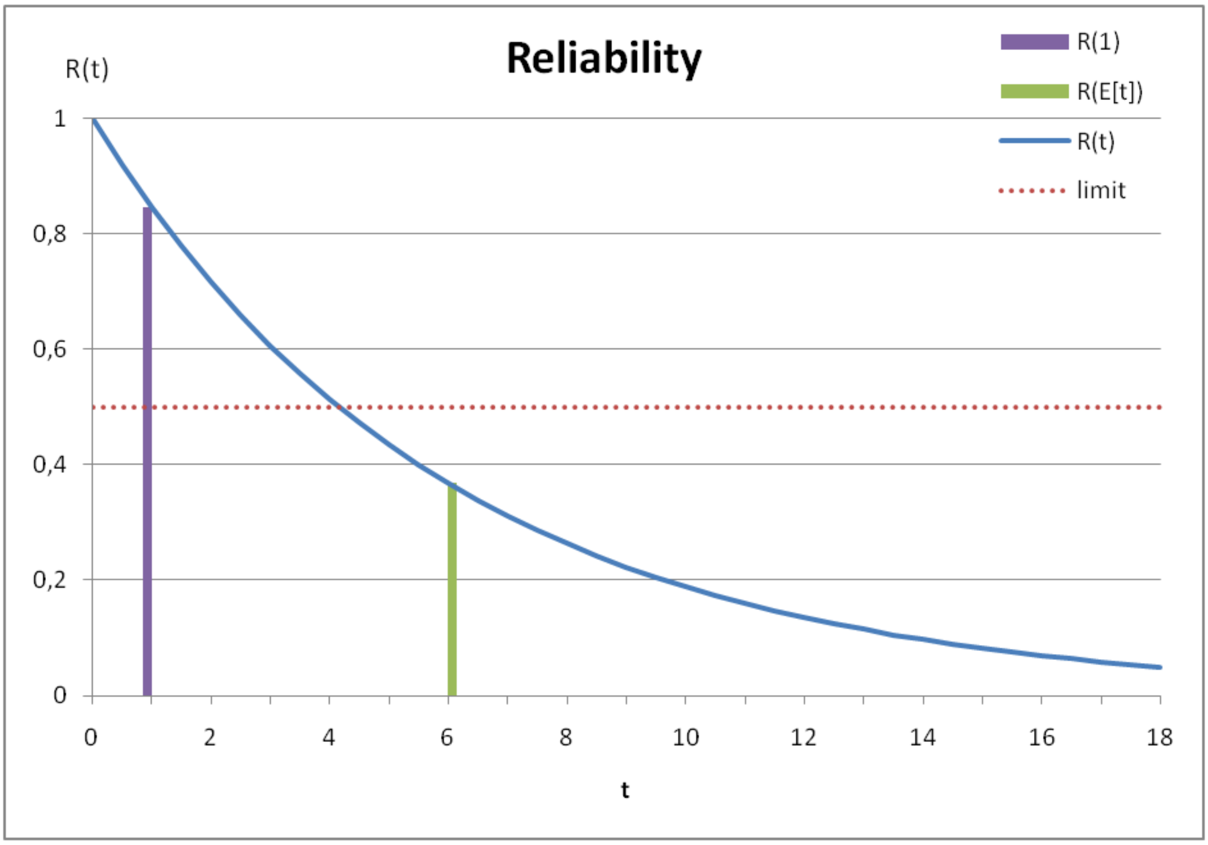


Figure 27: The reliability model with an average life-time of 6 years

The idea is to monitor the reliability of a device. If the value drops under a certain *limit*, in the example above 50%, the user will be notified to take further actions. In the diagram the monitored reliability $R(1)$ is at about 85%, so the probability of failure is rather small and therefore no user interaction is necessary. In the course of time the device becomes elder and so the average life-time is adjusted to the actual value. In this changed state the blue curve $R(t)$ will fall faster so that both calculated reliabilities ($R(1)$ and $R(E[t])$) will have an altered, lower value.

If necessary the user interaction for example can be to replace or maintain the device. In this case the risk management loop begins again with an updated average life-time which is adjusted to the current state of the resource.

3.2. Dynamic ERM – Duration model

When financial risks like a change of the interest rate are managed, a time and state continuous risk model is underlying. In special the duration is taken to model the interest risk of fixed income financial instruments (FIFI).

A fixed income financial instrument can be figured as a loan from a bank and is characterised by predetermined, regular, future returns over a set maturity. These returns are the future cash flows, which can consist partial of the interest payment and the redemption payment. The composition of these future cash flows distinguishes FIFIs in:

- Annuity loan – the periodic amount of the future cash flows is fixed. During the term the percentage of the interest payment becomes smaller, whereby the rate of the redemption payment is going up.
- Serial loan – the part of the redemption payment is equal throughout the term and the interest payments are going down.
- Bullet loan – the returns only consist of constant interest payments over the entire term. At the end one redemption payment has to be done, which is equal to the original investment.

Of course the effect of changing interest rates during the term is different between the three loan types, but the underlying risk model can manage the various possibilities by estimating the risk of the future cash flows.

If financial instruments and their changing interest rates are examined the questions about their risks will be:

- How much can the underlying risk differ?
- What is the impact of the change?

3.2.1. Risk Model

To answer these questions an adequate risk model for risk management for financial resources is needed as described in [14 p. 122ff.]. The volatility modelling is used to handle the stochastic of the risk factor for the interest risk. The generic risk model enables to calculate the maximal loss of the given event (LGE_t) at time t as described in (5).

$$LGE_t = Ex_t \cdot RL_t \cdot COR \tag{5}$$

The exposition (Ex_t) is the sum of the discounted future cash flows, the risk leverage (RL_t) is the modified duration, which itself is an extension of the Macaulay Duration. The cut off rate (COR) is evaluated by the standardised normal distribution with the given confidence coefficient and the predefined volatility of the interest rate.

3.2.2. Evaluation

For the evaluation of the FIFI the mathematical methods and formulas are based on [15 p. 70ff.].

fixed income financial instrument	t=0	t=1	t=2	time
NV_0	100,00			nominal value
R_0^N	4,00%			nominal interest rate
RP_t			100,00	redemption
$IP_t = NV_0 * R_0^N$		4,00	4,00	interest payment
$C_t = RP_t + IP_t$		4,00	104,00	future cash flow
Vola	1%			volatility
CC	95%			confidence coefficient
PE = 1 - CC	5%			probability of event
$Z \sim N^{-1}(95\%)$	1,645			z-value (quantile value)
ΔRF = Vola * Z	1,645%			cut off rate

Table 1: The initial input values for the LGE calculation

The fixed values for the computation are shown in Table 1. For the calculation example a FIFI with the maturity of two years has been chosen, the type of the FIFI is bullet loan, which can be seen by the unique redemption payment at the end. Because of its characteristic of the FIFI the future cash flows are finalised from the beginning.

The quantile value (*Z*) is computed by the standardised normal distribution and the chosen confidence coefficient (*CC*) as depicted in (6).

$$\begin{aligned} Z &\sim N(0,1) \\ &\sim N^{-1}(95\%) \\ &\approx 1{,}645 \end{aligned}$$

(6)

The probability of event (*PE*) is the probability with which the negative trend of the underlying risk factor will take place. The scale of the change is the volatility and the risk factor is the ratio for the case

- that the interest rate will fluctuate with 1%
- which can occur with the probability of 5%.

Formally the risk factor (ΔRF) is the product of volatility (*Vola*) and quantile value (*Z*) as shown in (7).

$$\begin{aligned} \Delta RF &= Vola \cdot Z \\ &= 1\% \cdot 1{,}645 \\ &\approx 1{,}645\% \end{aligned}$$

(7)

Table 2 illustrates the initial calculations. The interest rates ($R_{0,t}$) have been computed when the FIFI was created. This creation process has been worked out as part of the master thesis from Stefan Achleitner[16]. By discounting the future cash flows (C_t) the present value (PV_0) can be summed up.

Calculation in T_0:	t=0	t=1	t=2	time
$T_{0,t}$		1	2	maturity (in years)
$R_{0,t}$		1,38%	1,61%	interest rate
$DF_{0,t} = (1+R_{0,t})^{(-T_{0,t})}$		0,9864	0,9687	discount factor
$C_tPV_0 = C_t * DF_{0,t}$		3,9457	100,7403	initial valuation
$PV_0 = \Sigma\ C_tPV_0$	104,6860			initial valuation
$PV_0 = \Sigma\ C_t/(1+R_0^E)^{T_{0,t}}$	104,6839			
R_0^E	1,6016%			effective interest rate
$tC_tPV_0 = t*C_t/(1+R_0^E)^{(T_{0,t})}$		3,9369	201,4940	Macaulay duration
$D_{Mac,0} = \Sigma\ tC_tPV_0/PV_0$	1,9624			(in years)
$D_{mod,0} = D_{Mac,0}/(1+R_0^E)$	1,9314			modified Duration (in %)
$\Delta\%PV_0 = (-D_{mod,0})*\Delta RF$	-3,18%		relative	change of present value
$\mathbf{\Delta PV_0 = PV_0*\Delta\%PV_0}$	**-3,33**		absulute	

Table 2: Calculation at time T_0

The effective interest rate (R_0^E) is computed by the equation of the sum of the discounted cash flows and the mathematical formula of the present value as shown in (8). The equation is solved for the effective interest rate, whereby in ERPControl the resolution of this equation is done by parts of the Apache Math Commons library[6] and is explained in detail in [16 pp. 51-52].

$$\sum_0^T C_t \cdot DF_{0,T} = \sum_0^T \frac{C_t}{(1+R_0^E)^{T_{0,t}}}$$

(8)

The duration is a characteristic factor which informs about the elasticity of the FIFI focused on changes of the interest. In special the Macaulay duration ($D_{Mac,0}$) is a measurement for the average remaining time to maturity and is specified in years. As illustrated in (9) the modified duration ($D_{mod,0}$) extends the Macaulay duration and describes the relative change of the present value, if the effective interest rate varies with 1%.

[6] http://commons.apache.org/math/

$$D_{\text{mod},0} = \frac{D_{Mac,0}}{1+R_0^E}$$

(9)

The relative change of the present value ($\Delta\%PV_0$) can be calculated by a multiplication of the modified duration and the risk factor as explained in (10).

$$\Delta\%PV_0 = (-D_{\text{mod},0}) \cdot \Delta RF$$

(10)

To get the absolute value of the variation of the present value (ΔPV_0) in monetary units, the original exposition is multiplied with the relative change of the present value as shown in (11).

$$\Delta PV_0 = PV_0 \cdot \Delta\%PV_0$$

(11)

This result can also be evaluated by the formula of the generic risk model described in (5). The factors exposition, risk leverage and cut off rate are replaced by the factors present value, modified duration and risk factor as described in (12).

$$\begin{aligned} LGE_0 &= PV_0 \cdot (-D_{\text{mod},0}) \cdot \Delta RF \\ &= 104{,}686 \cdot (-1{,}9314) \cdot 1{,}645\% \\ &\approx -3{,}33 \end{aligned}$$

(12)

The result LGE_0 respectively ΔPV_0 is the maximal loss which could occur in 5% of the cases if the interest rate fluctuates with 1% in a negative way, or in other words 3,33 monetary units are the maximal loss which could happen under the given circumstances.

At the end of the first period a revaluation is done as shown in Table 3. Due to the fact that the first of the two years is over only the cash flow of the last, second period ($t=2$) will be considered in the revaluation. The future cash flow (C_2) is discounted and the actual effective interest rate (R_1^E) is calculated. Finally after both durations have been updated the new maximal loss (ΔPV_1) is the output of the revaluation process.

Calculation in T_1:	t=0	t=1	t=2	time
$T_{1,t}$			1	maturity (in years)
$R_{0,t}$			1,47%	interest rate
$DF_{0,t} = (1+R_{0,t})$^$(-T_{0,t})$			0,9855	discount factor
$C_tPV_0 = C_t * DF_{0,t}$			102,4933	initial valuation
$PV_1 = \Sigma\ C_tPV_1$		102,4933		initial valuation
$PV_0 = \Sigma\ C_t/(1+R_0^E)$^$T_{0,t}$		102,4934		
R_1^E		1,4700%		effective interest rate
$tC_tPV_1 = t*C_t/(1+R_1^E)$^$(T_{1,t})$			102,4934	Macaulay duration
$D_{Mac,1} = \Sigma\ tC_tPV_1/PV_1$		1,0000		(in years)
$D_{mod,1} = D_{Mac,1}/(1+R_1^E)$		0,9855		modified Duration (in %)
$\Delta\%PV_1 = (- D_{mod,1})*\Delta RF$		-1,62%	relative	change of present value
$\Delta PV_1 = PV_1*\Delta\%PV_1$		**-1,66**	absulute	

Table 3: Calculation at time T_1

At the end of the second period the maturity of the FIFI has ended, therefore no future cash flows are expected and no risk management - due to changing interest rates - is necessary.

4. Integration of ERM Reports in ERP-Control

To put the explained concepts from the Chapter 3 into practice, the procedures were implemented in ERPControl. At the beginning of the implementation for this master thesis, several students were involved as developers for different improvements in this ambitious software project. The code organisation and the code versioning were done by an installed subversion server at the department of Finance and Control at the Vienna University of Technology. The Eclipse platform[7] acts as the integrated development environment (IDE) for ERPControl, which facilitates the implementation by providing some helpful plug-ins for the JBoss Seam framework.

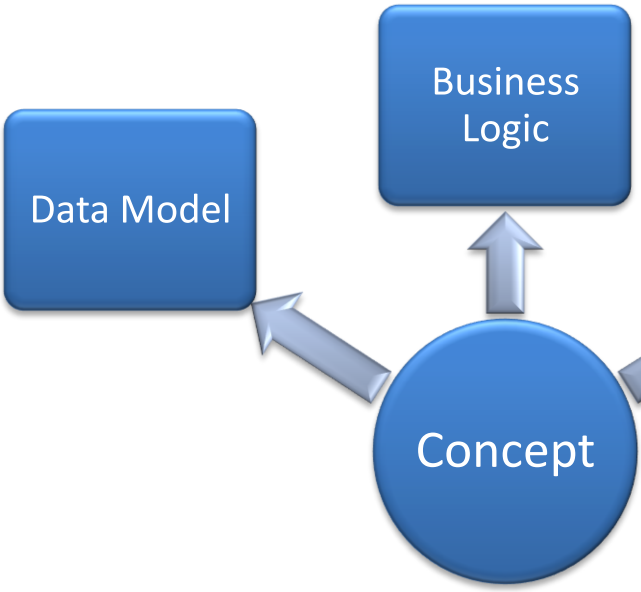


Figure 28: The concept leads to the implementation details

The evaluation concepts for ERM are the basis for the further development phase. As illustrated in Figure 28 the concepts define the different parts of the business model:

- Data Model – new business objects have to be created for the persistence of the input values, evaluation parameter and computed results. The entity beans are placed in the `at.ac.vut.bpmis.bo.erm` package.

[7] http://www.eclipse.org/

- Business Logic – the business activities, which provide the functionality for the desired calculations, are centralised in a session bean and its interface, which are packed in `at.ac.vut.bpmis.jbpmservices.dcps.erm`.
- Presentation – the business processes define the way to compute the results and to display the reports for ERM. Presentation includes the jBPM business processes, page flows, managing session beans and JSF, which generates the GUI. These components are at different positions in the software project. All jBPM relevant parts can be found in the `EarContent` folder, the session beans, which are necessary for the management functionality of the jBPM business processes, are located in the `at.ac.vut.bpmis.jbpmorchestration` package and finally all files, which belong to the GUI, are placed in subfolders of `WebContent` (`/core/operations/erm` and `/core/report/erm`).

Document Structure Hint

To provide a complete reference of all implementation details, the risk models are documented with the same structure like the general system implementation. Till to a certain extend both risk models operate in the same way:

- The persistence is done by the same EBs.
- The business logic is placed in the same SB.
- The business process requires the same components: process definition and process manager.

Of course the evaluation of the risk events makes a distinction between the limit types, but the way to get the result is the same. The greatest difference can be found in the GUI, because of the varying required input values. Therefore if both risk models can be explained in the same way, it will be referenced to Chapter 4.1. Otherwise the specialities of each risk model are described in the according Chapters 4.2 respectively 4.3.

4.1. Integrating ERM – general System Implementation

4.1.1. Business Objects: defining the Persistence

Packages: `at.ac.vut.bpmis.bo.erm; at.ac.vut.bpmis.bo.planning;`

To manage various risks the existing database model was extended to record the different aspects for each risk model. The nested structure of the entity beans, which were created to record the risk events and their values, can be seen in Figure 29.

Beginning point of the structure for ERM is the biggest ball in the Venn diagram, which illustrates the resources. An economic resource is always the start for risk management, because a special risk of a single resource shall be managed. The next smaller ball illustrates the risk category to which the monitored risk belongs to. After choosing the risk category the associated risk models will be presented, whereupon an according one has to be selected. The risk model specifies the evaluation and provides the structure to persist the model parameter. Finally the risk event is created which is the entity bean with the input values and the calculation results as object properties.

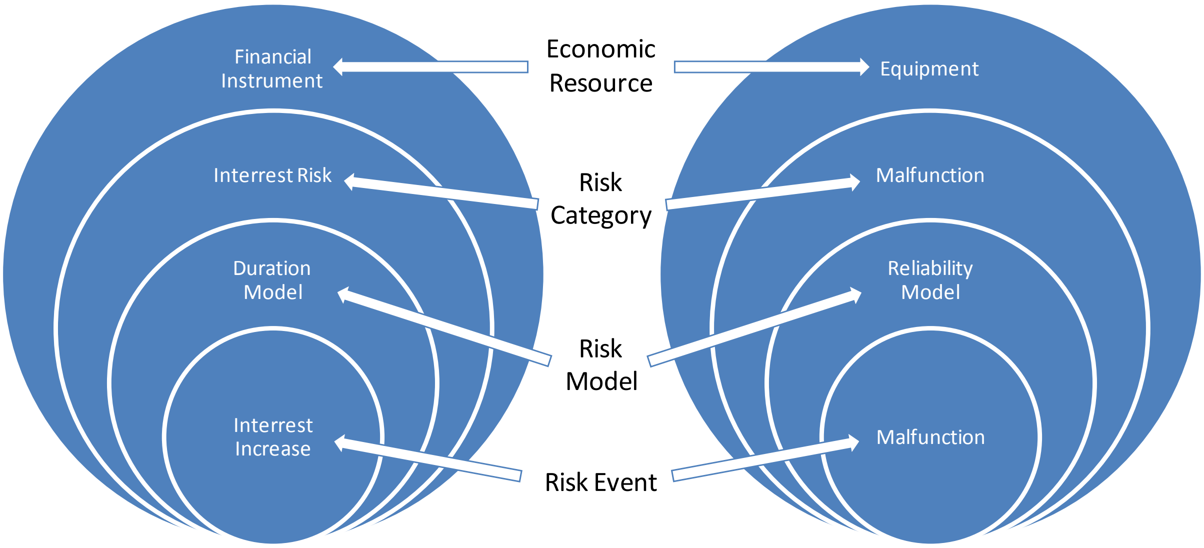


Figure 29: The nested structure of the risk management business objects

The database model in Figure 30 presents the UML class diagram with the mentioned entity beans. The diagram also clarifies the associations among the entities, which are in this case bidirectional and unidirectional compositions and multi-associations. The entity bean `ValueEntity` is located in the `at.ac.vut.bpmis.bo.planning` package, because it is also used to store the values of planning events, which has been implemented by Daniel Fellner [17]. The entity bean `EconomicResource` is part of the existing core structure of ERPControl and is located in the `at.ac.vut.bpmis.bo.rea` package.

A composition is a strong connection between two entities and is represented by a line with a solid diamond shape. A composite aggregation is a special case of an aggregation and describes a connection between a whole and its parts. The speciality is that the whole may exist without its parts, but the inverse way is not possible. So an `EconomicResource` can have none or more `RiskEvent`s, but there will not be a `RiskEvent` without the connection to an `EconomicResource`. If an instance of the container class, the `EconomicResource`, is deleted, all the instances of the contained class, the `RiskEvent`, will be deleted too.

The other type is the association with a multiplicity of 0..* in both directions, which can be found between the `RiskEvent` class with the `RiskEventCategory` class. This association specifies that an instance of a `RiskEvent` class can belong to zero or many instances of the `RiskEventCategory` class. The same logic is valid in the contrary direction. The association is named “`categorises`”, what denotes the meaning of this connection.

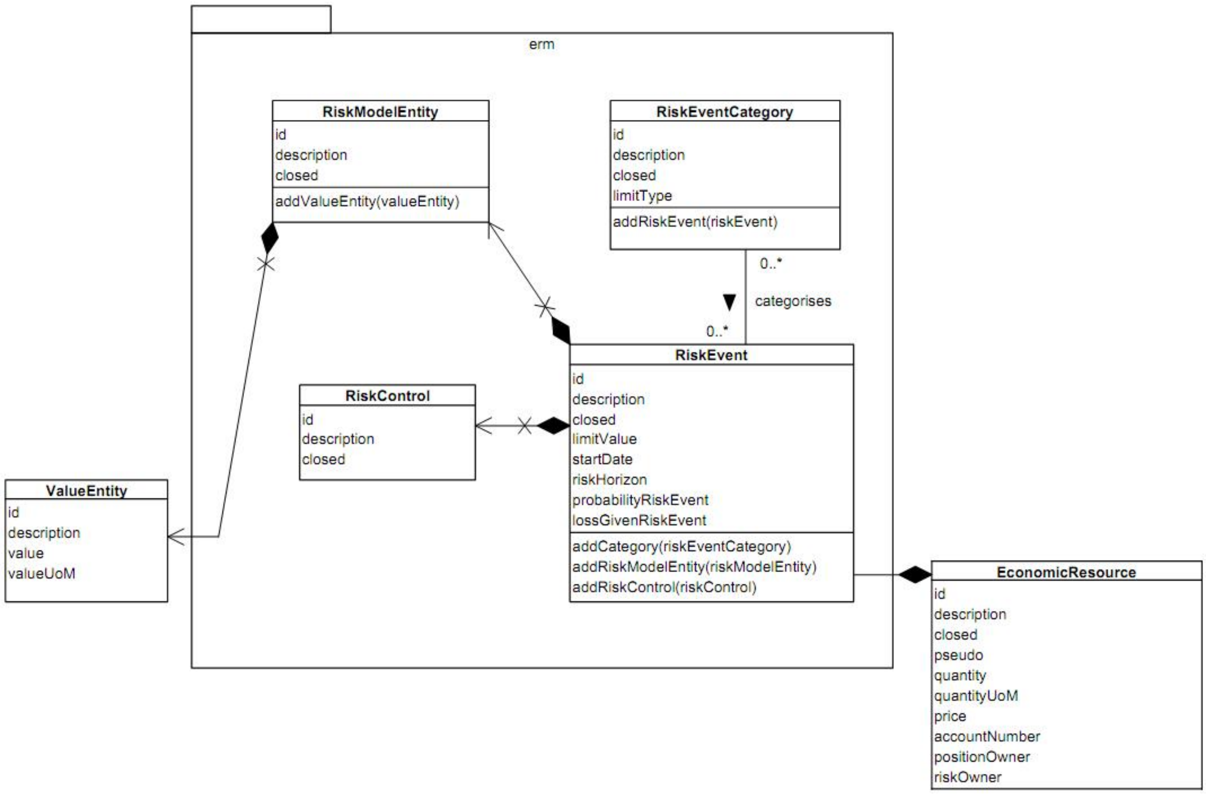


Figure 30: ERM database model, the erm-package extends the existing schema

An association or composition can be uni- or bidirectional. Unidirectional connections can only be navigated as defined by the arrow. A `RiskModel` entity can fetch its associated `ValueEntity` objects from the database, but a `ValueEntity` object cannot navigate to the linked `RiskModel` entity, because it is not necessary and therefore not implemented.

4.1.2. Business Activities: providing the Functionality

Package: `at.ac.vut.bpmis.jbpmservices.dcps.erm;`

The business activities are packed into a single stateless session bean to have one object, which contains the procedures for the evaluation of the risk events. The advantage of a stateless SB compared to stateful SB is that the SLSB can be invoked much faster, because the application server needs not to save its state in the memory.

Code 9: Interface for risk evaluation

```
package at.ac.vut.bpmis.jbpmservices.dcps.erm;
import ...;

public interface RiskEvaluation {

    public double evaluateReliability(RiskEvent riskEvent) throws
Exception;
```

```
    public double evaluateReliability(EconomicResource equipment)
throws Exception;

    public double evaluateLGRE(RiskEvent riskEvent) throws Exception;

    public double evaluateLGRE(EconomicResource financialInstrument)
throws Exception;
}
```

The package includes two classes: the `RiskEvaluation` class is the interface with the provided business methods. These methods can be invoked by other session beans or via the GUI. As shown in Code 9 the interface specifies the method´s parameter, the return value and the exception type.

The second class is called `RiskEvaluationBean` and implements the interface class. The SB class contains the actual functionality and has to implement all business methods as defined in the interface, but these can be extended with additional methods if required.

Code 10: SLSB for risk evaluation

```
package at.ac.vut.bpmis.jbpmservices.dcps.erm;
import ...;

@Stateless
@Name("riskEvaluation")
public class RiskEvaluationBean implements RiskEvaluation {

    @EJB
    private IPricingCalculationSerialBullet pricingCalculation;

    ...
}
```

In Code 10 an extract of the SLSB `RiskEvaluationBean` can be seen. The annotation `@Stateless` instructs the application server to provide the SLSB if required. The invocation is done by calling `"riskEvaluation"`, the available methods are denoted in the public interface and will be explained in detail later. A speciality is that another session bean is injected. The `@EJB` annotation marks a required interface, which provides a business method which is needed for the computation of the result in the `RiskEvaluationBean`.

4.1.3. Business Processes: organising the Presentation

Packages: `at.ac.vut.bpmis.jbpmorchestration; EarContent; WebContent;`

The organisation of the business processes requires different components:

- The tasks during the process are provided by a process manager SB, which has the necessary procedures implemented.

- The modelled jBPM business specifies the orchestration of the various tasks within the process.
- The correct page navigation is defined as page flows, which guide the user through the GUI and build atomic, prime units, which are called conversations.
- The GUI consists of many webpages, which present the data or provide the input fields.

Process Manager

A SFSB provides the components and methods, which are required during a process. Analogous to the business activity SFSB that provides the evaluation procedures, this process manager includes several annotated attributes with components for the presentation, input processing and state management. So the difference is that the process manager does not evaluate the result, but it just prepares the given data, manages the values for the process and handles the state during the process steps.

Code 11: A SFSB acts as process manager for the risk creation process

```
package at.ac.vut.bpmis.jbpmorchestration;
import ...

@Stateful
@Name("riskEventManager")
public class RiskEventManager implements IRiskEventManager {

    ...

    @EJB
    private RiskEvaluation riskEvaluation;

    @DataModel
    private List<RiskEventCategory> riskEventCategoryList;

    @DataModelSelection(value="riskEventCategoryList")
    private RiskEventCategory riskEventCategorySelected;

    @Out(scope=ScopeType.BUSINESS_PROCESS, required=false)
    @In(scope=ScopeType.BUSINESS_PROCESS, required=false)
    private Long riskEventCategoryId;

    ...

    @Factory("riskEventCategoryList")
    public void displayRiskEventCategories() {
        riskEventCategoryList = entityManager.createQuery("from
    RiskEventCategory rec where rec.closed = false").getResultList();
    }
```

```
    ...
}
```

The SFSB "`riskEventManager`" is illustrated in Code 11 and implements the interface `IRiskEventManager`, which specifies the according business methods. The `@EJB` annotated `riskEvaluation` component is the SFSB, which was explained above and provides the functionality to evaluate the result. The other SFSB properties are either for data presentation (e.g. specified by the `@DataModel` and `@DataModelSelection` annotation), for data preparation (e.g. the `@Factory` annotated method fetches EB from the database) or for state management of the business process (e.g. persisting an `ID` by the `@In` and `@Out` annotations).

Code 12: Code extract for the final process step risk evaluation

```
@StartTask
@Begin(join=true, pageflow="pf_bp_riskEvent")
public String beginValueEvaluation() {
    riskEvent = entityManager.find(RiskEvent.class, riskEventId);
    return "addRE_pf_value-evaluation";
}

public void evaluateRiskEvent() {

    try {
        if(limitType == null || limitType.equals(""))
            limitType = riskEvent.getLimitType();

        if(limitType.equals("Reliability"))
            result = riskEvaluation.evaluateReliability(riskEvent);
        else if(limitType.equals("Maximal loss"))
            result = riskEvaluation.evaluateLGRE(riskEvent);

        showResult = true;

    } catch (Exception e) {

        showResult = false;
    }
}

@EndTask(transition="next")
public void endValueEvaluation() {
    riskEvent.setClosed(false);
    entityManager.persist(riskEvent);
}
```

The part of the SFSB "riskEventManager", which is presented in Code 12, manages the last step of the business process for adding a risk event. The first method fetches the EB riskEvent from the database and starts the task-node, which is defined by the return-value.

The second method will be called, if the user wants to process the result of the risk event. Then the SFSB riskEvaluation does the actual result calculation. The computation is executed according to the limitType of the current riskEvent, so either the reliability or the maximal loss of the given riskEvent is returned.

The last method finishes the task-node in the business process. After the persisting of the manipulated riskEvent, the transition "next" is invoked by the @EndTask annotation, which assigns the jBPM to forward the next task-node as defined in the process definition.

jBPM Business Process

A jBPM business process instructs the installed Seam component in the application server, which action has to be performed if a process is initialised respectively during the single process steps. As illustrated in Code 13, the XML file starts with an XML definition. The name of the process is defined by the root-element process-definition. This root-element includes the tags start-state, end-state and the intermediate task-nodes. There must be exact one start-state element, which instructs jBPM with which task-node the process is stated.

Several task-nodes are specified that guide the user through the necessary activities. A task-node includes the task, which have to be executed, and the transition to the next task-node. Depending on the result of the business activity, different transitions can be executed. In the example in task-node "category-selection", one of three transitions can be taken. If the current task-node is aborted, the business activity with the "abort" transition will be invoked and the process will wait again at the current task-node. If the business activity is annotated with "next", the transition to the next, logical task-node "resource-selection" is invoked. If the user cancels the process, the according business activity is marked with the value "cancel" and the process will move to the end-state, which determines the whole jBPM business process.

Code 13: jBPM business process for creating a risk event

```
<?xml version="1.0" encoding="UTF-8"?>
<process-definition name="bp_riskEventAdd">

    <start-state name="start">
        <transition to="category-selection"></transition>
    </start-state>
```

```
    <task-node name="category-selection">
        <task name="reAdd_category-selection">
            <description>
               risk event category selection
            </description>
            <assignment actor-id="#{actor.id}"/>
        </task>

        <transition to="category-selection" name="abort"></transition>
        <transition to="resource-selection" name="next"></transition>
        <transition to="end" name="cancel"></transition>
    </task-node>

    <task-node name="resource-selection">
        ...
    </task-node>

    ...

    <end-state name="end"></end-state>

</process-definition>
```

The process can be modelled with the JBoss jBPM Graphical Process Designer (GPD)[8], which is an Eclipse plug-in. The GPD can create the XML code out of the graphical designed process and contrary, the hardcoded process can be illustrated as depicted in Figure 31. The graphic shows the `start-state`, the `end-state`, the `task-node`s and the available `transition`s of the business process "`bp_riskEventAdd`". The process has to follow the `transition`s between the `task-node`s, another way is not possible.

[8] http://www.jboss.org/jbpm/

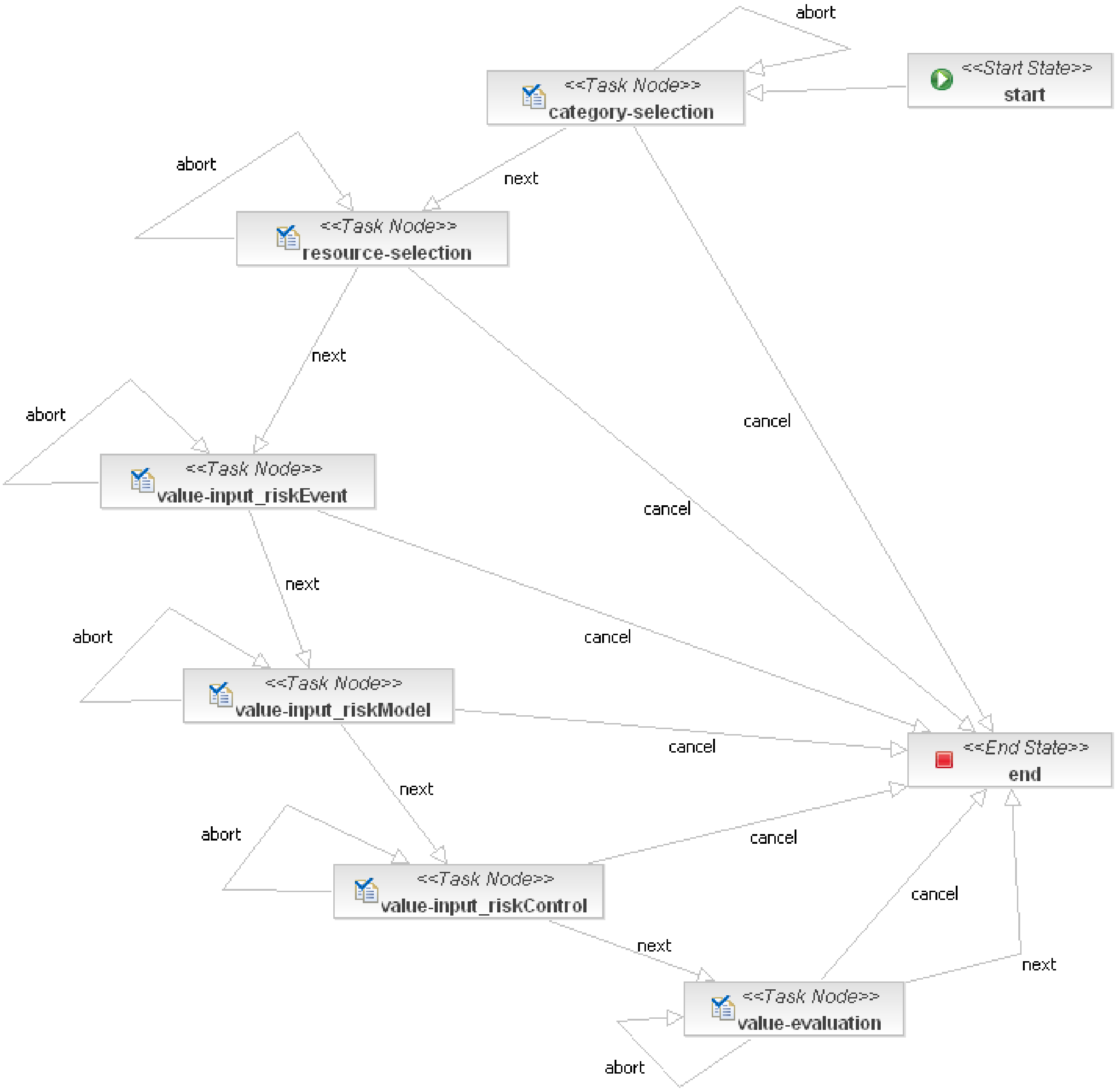


Figure 31: Process image for creating a risk event

Pageflow

If the business process stands in a certain process step, then a pageflow will specify the necessary webpages, which have to be invoked. During a pageflow, the user is within a conversation, which represents a non-separable unit of steps. Only if the whole conversation has been finished successfully, the according task-node in the business process changes to the "next" transition.

Code 14: jBPM page flow definition the risk event creation process

```
<pageflow-definition ... name="pf_bp_riskEvent">
    <start-state name="start">
        <transition ...

        <transition name="addRE_pf_value-evaluation"
        to="value-evaluation" />

    </start-state>

    <page ...

    <page name="value-evaluation"
    view-id="/core/operations/erm/addingValueEvaluation.xhtml">
        <redirect />

        <transition name="next" to="end">
            <action
            expression="#{riskEventManager.endValueEvaluation()}" />
        </transition>

        <transition name="abort" to="end">
            <action expression="#{riskEventManager.addAbort()}" />
        </transition>
    </page>

    <page name="end"
    view-id="/core/operations/erm/addingOverview.xhtml">
        <redirect />
    </page>
</pageflow-definition>
```

In the extract of the pageflow definition "pf_bp_riskEvent" the interesting part for the value evaluation can be seen in Code 14. The invocation of the beginValueEvaluation() method starts the pageflow, the return-value specifies, which transition of the start-state shall be taken. In the case of the final process step for the risk evaluation, the pageflow goes to the page element "value-evaluation". The view-id instructs jBPM, which webpage has to be presented in the GUI. When the user clicks the command button "Enter" on the webpage, the button action "next" specifies to take the first transition in the page element. The action denotes the method riskEventManager.endValueEvaluation() of the process manager, which executes the finalising procedures to end the process step. The transition directs the pageflow to the page element "end" and the webpage for the process overview will be displayed in the GUI.

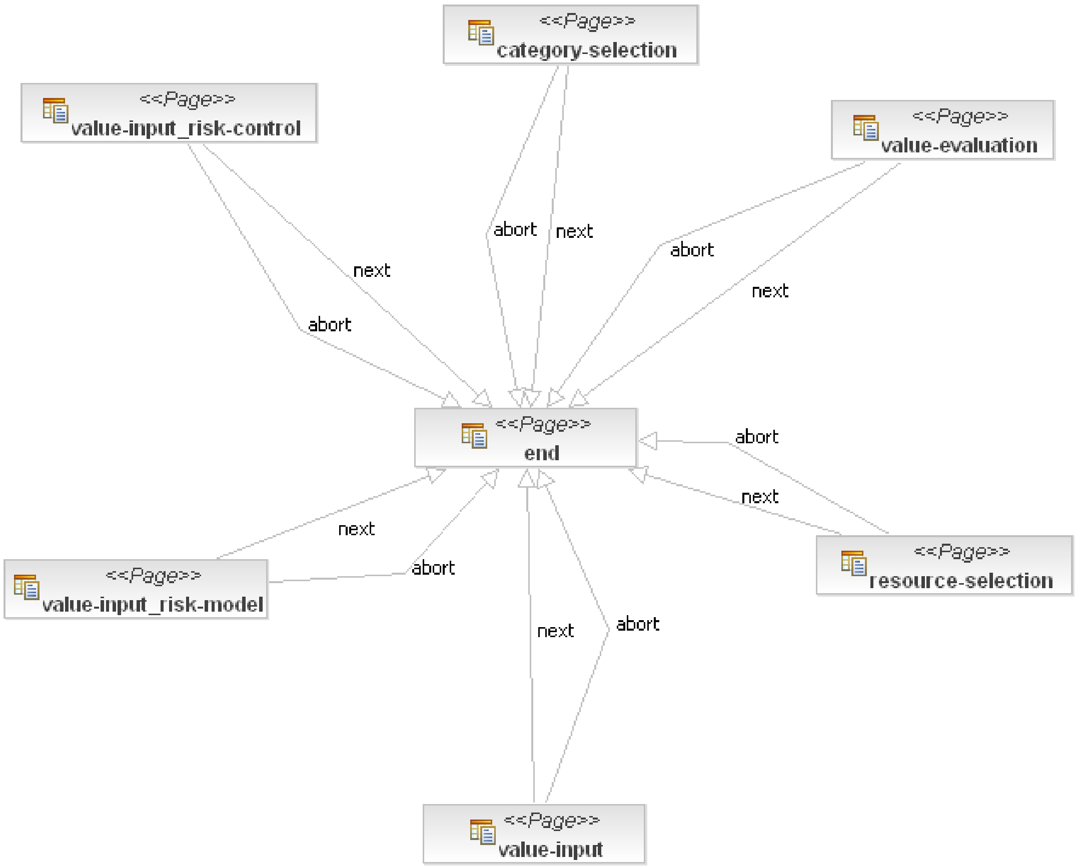


Figure 32: Pageflow illustration with the available pages and transitions

Due to a malfunction of the GPD in the diagram module the `start-state` is missing in Figure 32. For normal the graphic should be alike the image of the business process, at which the `start-state` is the initial point of the process. Even if the diagram is incomplete, the processing of the pageflow works fine. The available transitions between the `page` elements and the `page "end"` can be seen.

4.1.4. Dynamic Webpages: constructing the User Interface

The webpages are the only parts of the explained techniques in this Chapter, which the user really sees and interacts with. The other parts are hidden from the user by the application server. The structure of the GUI is constructed through the JSF framework and the RichFaces extension. Templates build the skeletal structure of the webpages, for example the layout, the menu, etc. as explained in Chapter 2.4.1.

The processes for ERM in ERPControl can be found in the "Reporting" and in the "Analytical Planning/Design" menu as shown in Figure 33, in which the page for "Risk Planning" is active.

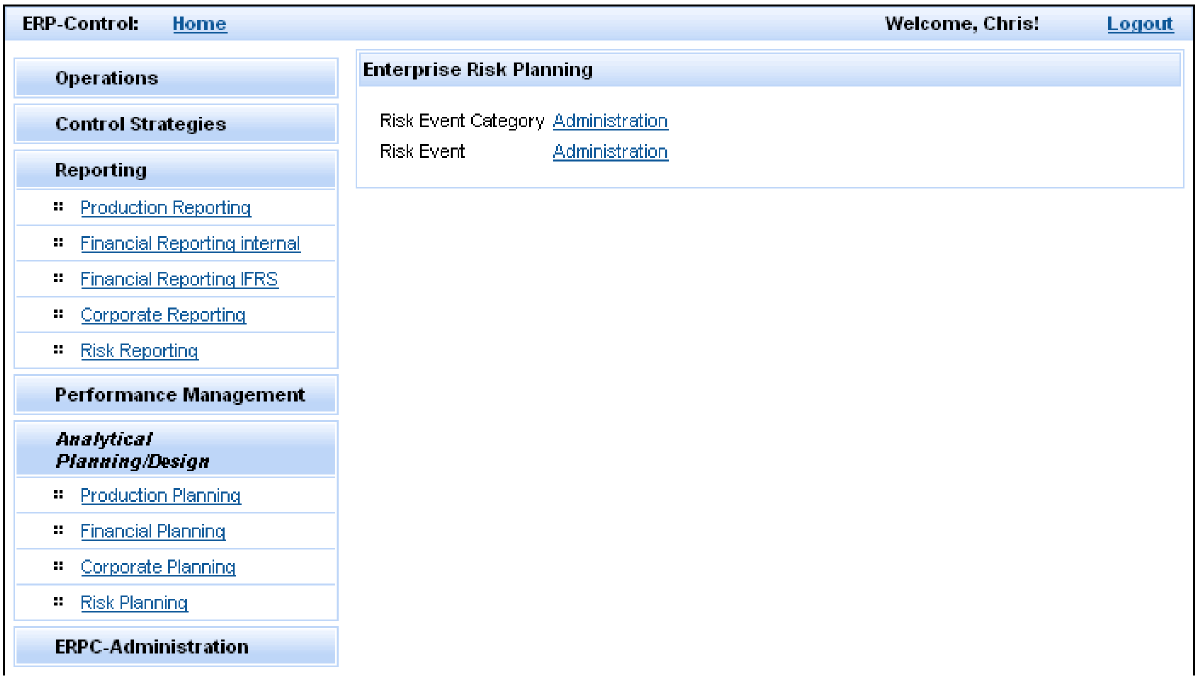


Figure 33: ERM in ERPControl

In order to simplify matters, the further illustrations will only show the content part of the ERPControl screenshots.

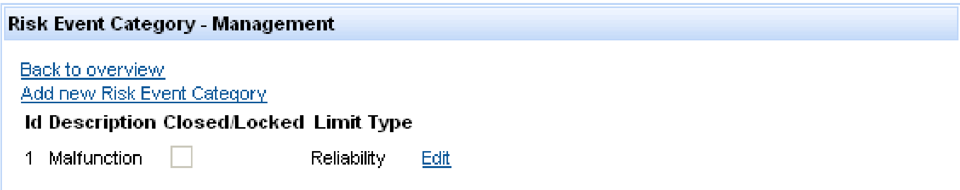


Figure 34: Overview for risk event category management

To evaluate risk events, several preparation tasks have to be done before. At first the risk event categories have to be created. Figure 34 depicts the screen with the overview of the current available risk event category, which is in this case “Malfunction”.

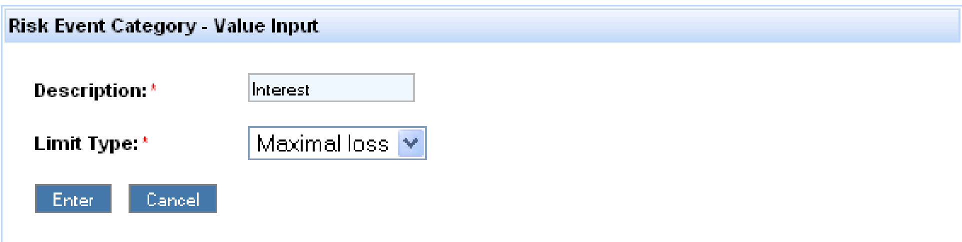


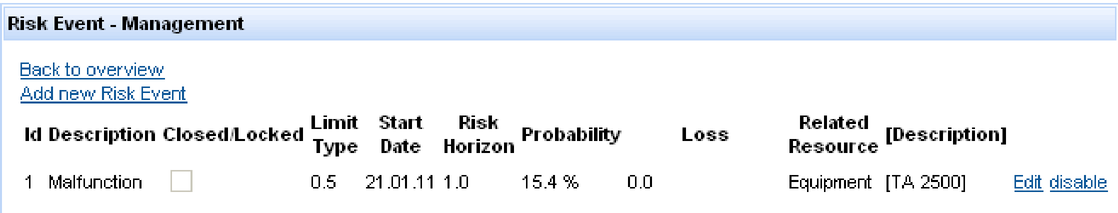
Figure 35: Adding a new risk event category

To add a new risk event category, a description and the according limit type have to be denoted as shown in Figure 35.

The risk event category with its properties specifies the limit type and therefore the risk model for the future risk events. Current implemented limit types are “Maximal loss” and “Reliability”, which are explained in detail in Chapter 4.2 and 4.3.

Figure 36 illustrates the overview of the presently available risk events. The important properties of a risk event can be found in the columns, at the end of each row the according commands are located. Action “Edit” initialises a pageflow to manipulate the chosen risk event.

An entity, which has been already used in the system, must not be deleted, otherwise the missing reference causes an error in the system. Due to this fact a risk event must not be removed from the database, but it can be disabled, which instructs the system to do not use this entity anymore, but it has the advantage that the reference still works.



Risk Event - Management

Back to overview
Add new Risk Event

Id	Description	Closed/Locked	Limit Type	Start Date	Risk Horizon	Probability	Loss	Related Resource	[Description]	
1	Malfunction	☐	0.5	21.01.11	1.0	15.4 %	0.0	Equipment	[TA 2500]	Edit disable

Figure 36: Overview for risk event management

Figure 37 presents the overview of the business process for adding and evaluating a new risk event. Six process steps have to be executed to create a new risk event. Furthermore a new business process instance has already been initialised and is waiting in the first task-node for user interaction. The action link starts the risk event category selection, in which one of the available risk event categories can be chosen. After the successful execution, the business process instance changes into the second process step.

In the second task-node the desired economic resource has to be selected. The user has to choose the resource according to the elected risk event category. Of course the user will be notified if the resource does not match to the limit type, e.g. the (technical) reliability cannot be estimated for a financial instrument.

Up to the second process step, the tasks for both risk models have been the same. From the third task-node on, it is distinguished between reliability and maximal loss as explained in Chapter 4.2.4 and 4.3.4.

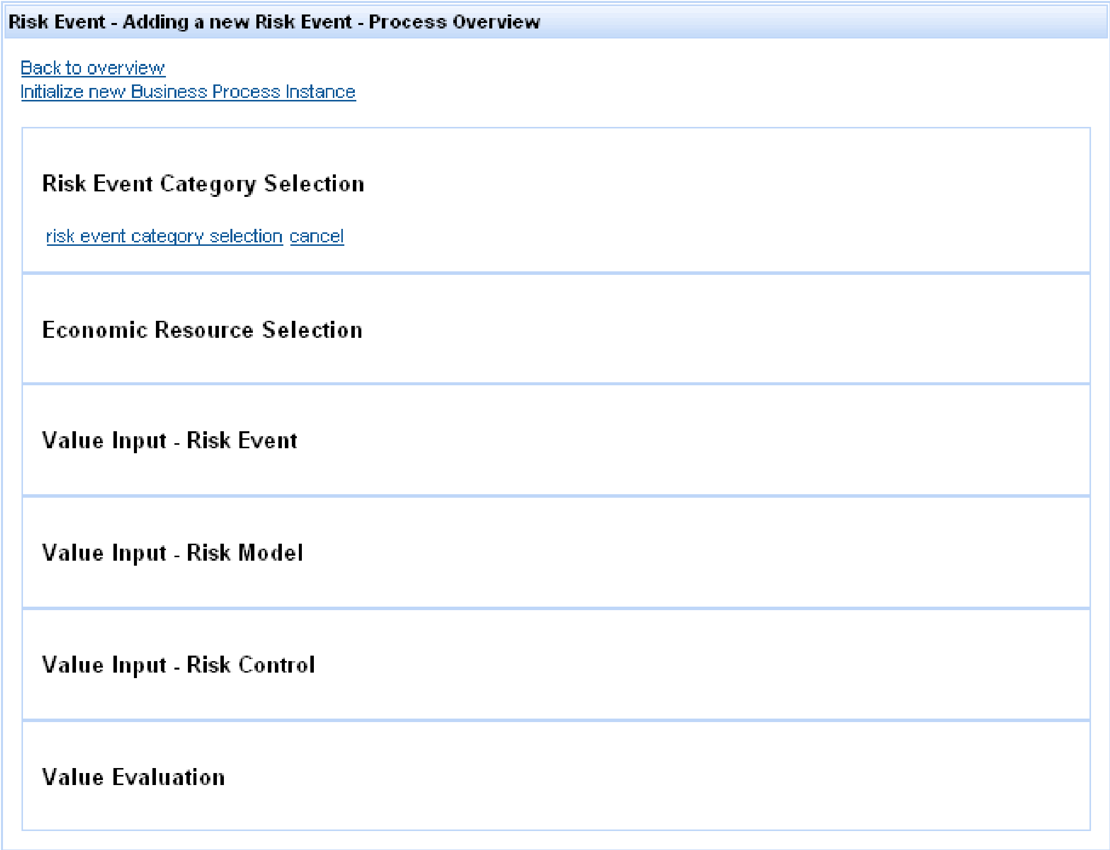


Figure 37: Business process overview which an active instance for adding a new risk event

4.2. Integrated ERM – Reliability

The implementation details for evaluation the reliability of devices are explained in the following pages, from the underlying database structure to the GUI.

4.2.1. Entity Beans: specifying the Data Structure

The according entities and the database structure for the reliability risk model are described in Chapter 4.1.1.

The EBs record the following input values, which are necessary to construct the risk model and to evaluate the reliability for a device:

- The risk event category
  - the risk event is categorised for risk model specification, reporting, etc.
  - is linked with the `RiskEvent` by a bidirectional association
- The economic resource
  - the corresponding equipment, for which the reliability shall be estimated

- is connected with the `RiskEvent` through a composition
- The limit
  - the critical value which causes the alert if the reliability is under this boundary
  - is recorded as a property of the `RiskEvent`
- The start date
  - the point in time at which the equipment was purchased of maintained
  - is recorded as a property of the `RiskEvent`
- The risk horizon
  - defines the time span for which the reliability will be processed
  - is recorded as a property of the `RiskEvent`
- The average life-time
  - the appreciated durability of the corresponding equipment
  - persisted as a `ValueEntity` of the `RiskModel`
- The risk control
  - denotes the action, which shall be taken in case of exceeding the limit
  - is linked to the `RiskEvent` through a compositon

These user inputs are entered during the business process, the according screenshots can be seen in Chapter 4.2.4.

4.2.2. Session Beans: implementing the Business Logic

After the user inputs have been persisted into the database and the user started the risk evaluation, the SFSB `RiskEventManager` invokes the according method of the SLSB `RiskEvaluationBean`.

Code 15: Methods for evaluating the reliability

```
/**
 * Method for evaluating the reliability of the given RiskEvent.
 * All necessary entities are fetched from the DB and checked.
 * The actual values are computed.
 * All computed values are persisted in the according BO.
 * Persists the pRE in the RiskEvent entity.

 * @param riskEvent the RiskEvent with the necessary values
 * @return the reliability R[t]
 * @throws Exception in case of wrong/missing values
```

```
 * @see RiskEvent
 * @see RiskModelEntity
 * @see ValueEntity
 */
public double evaluateReliability(RiskEvent riskEvent) throws Exception
{ ... }

/**
 * Method for evaluating the reliability of a given Equipment.
 * details: {@link RiskEvaluation#evaluateReliability(RiskEvent)}

 * @param equipment the EconomicResource must be an instance of
    Equipment with the associated RiskEvent
 * @return the reliability R[t]
 * @throws Exception in case of wrong/missing values
 */
public double evaluateReliability(EconomicResource equipment) throws
Exception { ... }
```

Code 15 represents the extract of the SLSB `RiskEvaluationBean`, which will be invoked, if the user asks for the result of the reliability. The invocation can be executed in two ways, whereby the difference lies in the passed parameter of the methods. Either the equipment as subtype of the class `EconomicResource` is given over, or the `RiskEvent` itself. Both ways are possible and lead to the same result, the current reliability of the device.

The reliability is computed according to the described concept in Chapter 3.1. The detailed code documentation can be found in the Javadoc[9]. This documentation is generated out of the Java code comments and technically describes the classes, methods and parameter in detail.

As an example Figure 38 illustrates the Javadoc page of the SLSB `RiskEvaluationBean`. After the short overview at the beginning, the part of the class definition describes the meaning of the class, the implemented interfaces and shows in this case the SB annotations. The constructor and the method summary can be found in the following tables. The biggest part of the Javadoc page is not shown in this extract, it is the detailed description of the constructor and all methods. A really useful feature is the linking of mentioned classes to their own Javadoc page, which are the blue coloured hyperlinks. But the hyperlinks may not only refer to local Javadoc pages, it is also possible to link external Javadoc webpages. This speeds up the navigation between the different classes and simplifies the development of extensive structured systems like ERPControl.

[9] Javadoc in Java 6: http://download.oracle.com/javase/6/docs/technotes/guides/javadoc/index.html

at.ac.vut.bpmis.jbpmservices.dcps.erm

Class RiskEvaluationBean

```
java.lang.Object
  └ at.ac.vut.bpmis.jbpmservices.dcps.erm.RiskEvaluationBean
```

All Implemented Interfaces:

RiskEvaluation

```
@Name(value="riskEvaluation")
public class RiskEvaluationBean
extends Object
implements RiskEvaluation
```

The SLSB RiskEvaluationBean.
This stateless session bean provides the functionality to evaluate the result of a RiskEvent. It implements the public interface RiskEvaluation.

Author:

Christoph Rodler

Constructor Summary
`RiskEvaluationBean()` The default constructor.

Method Summary	
double	`evaluateLGRE(EconomicResource financialInstrument)` Method for evaluating the possible loss of a given FinancialInstrument.
double	`evaluateLGRE(RiskEvent riskEvent)` Method for evaluating the possible loss of a given RiskEvent.
double	`evaluateReliability(EconomicResource equipment)` Method for evaluating the reliability of a given Equipment.
double	`evaluateReliability(RiskEvent riskEvent)` Method for evaluating the reliability of the given RiskEvent.

Figure 38: Extract of the generated Javadoc

4.2.3. jBPM: orchestrating the Business Processes

jBPM Business Process and Pageflow

The uniform jBPM business process and the corresponding pageflow for both risk models are specified in Chapter 4.1.3.

Process Manager

The process manager itself is also the same for both risk models, but it distinguishes due to the limit type, which inputs fields have to be created and which evaluation method is taken.

Code 16: Distinction in the process manager´s evaluation method because of the limit type

```
if(limitType.equals("Reliability"))
     result = riskEvaluation.evaluateReliability(riskEvent);

else if(limitType.equals("Maximal loss"))
     result = riskEvaluation.evaluateLGRE(riskEvent);
```

As presented in Code 16, the limit type is the determining factor, which evaluation method of the SLSB `RiskEvaluationBean` is taken. The following processing of the result is done in the same way for both risk models.

4.2.4. JSF: presenting the Graphical User Interface

As shown in Chapter 4.1.4, the jBPM business process has not differentiated between both risk models before the third process step.

When the user starts the third task-node, the value input page for the reliability model will be invoked. Figure 39 depicts the values limit, start date and risk horizons, which are required fields for the user, what is indicated through the red asterisk after the label. For convenience a date assistant is provided next to the date input field. Next to the other input fields some information about the input value is denoted.

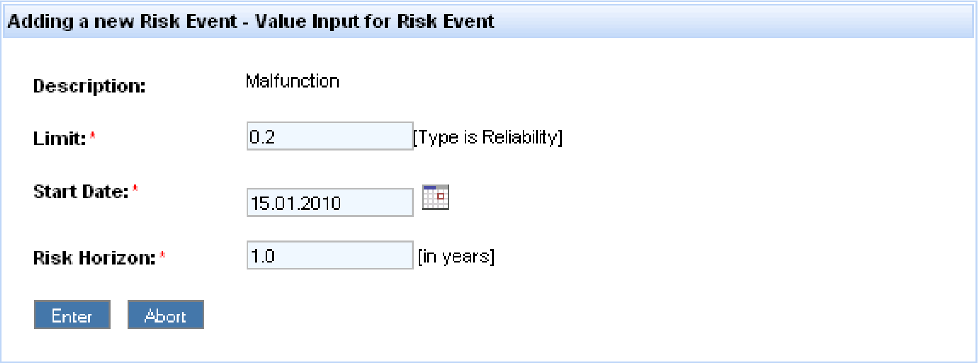


Figure 39: Value input page for risk event of type reliability

The same value input page is presented in Figure 40, but with a validation error for each input field. If the user enters an invalid value, the validation is executed immediately through AJAX functionality. The user will be notified just after the input field left the focus and an appropriate error message is displayed next to it, if an error occurs.

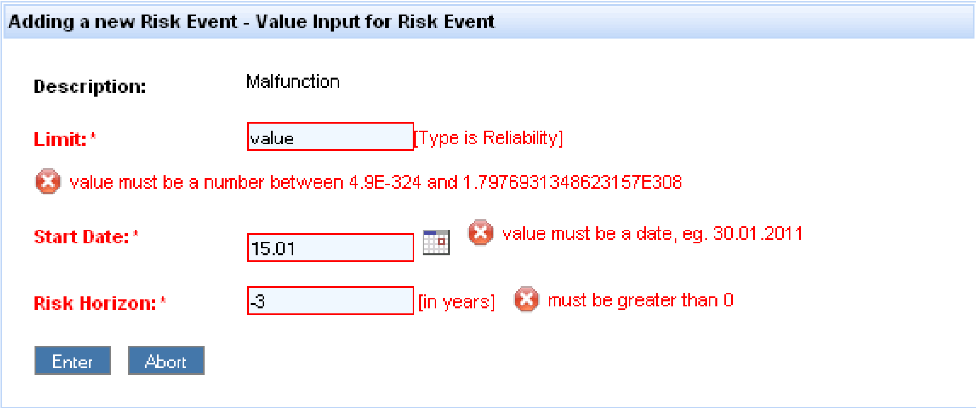


Figure 40: Value input page with validation errors

After the user has corrected the values, the validation is done again and if it is successful, the validation error will disappear.

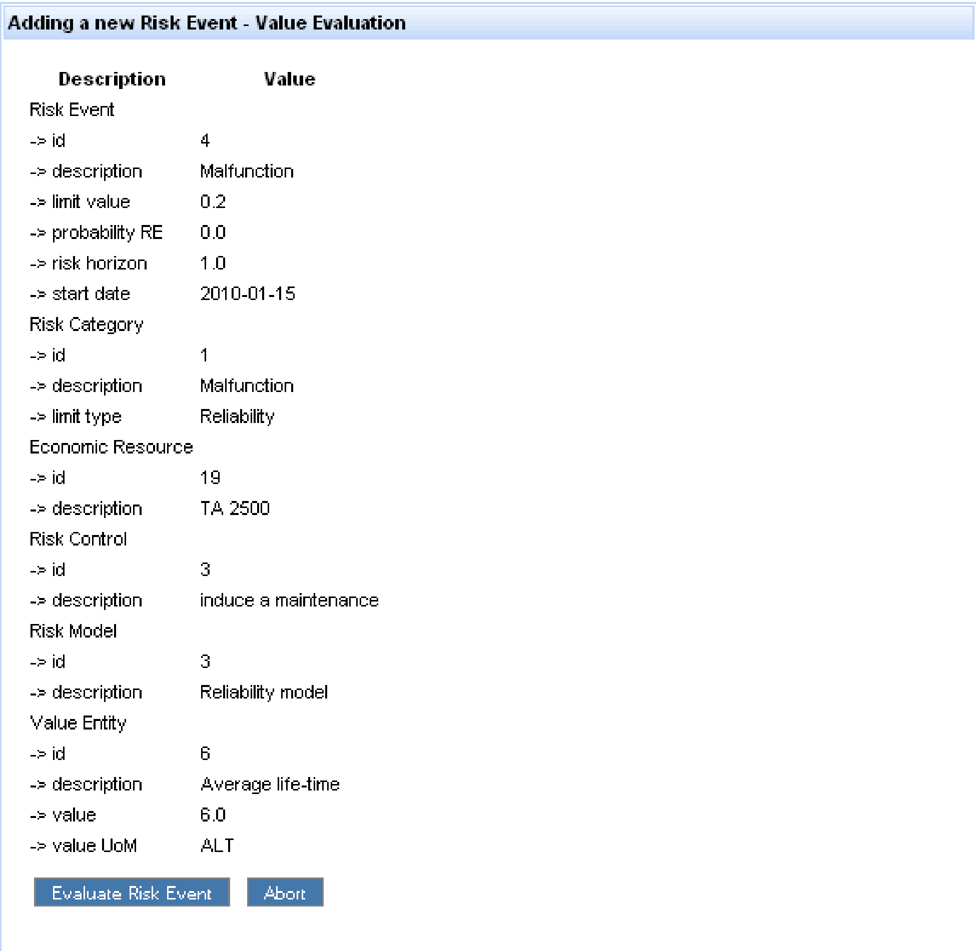


Figure 41: Summary of the input values for reliability evaluation

In the next task-node, the input for the risk model has to be specified. In the case of the reliability model the average life-time in years is required.

In the value input page, which is process step number four, an adequate description for the risk control has to be entered. The risk control describes, which action has to be performed if the limit of the risk event has been exceeded.

The last task-node performs the risk evaluation in two sub-tasks. First, the entered input values are summarised for a review before the evaluation is executed. The corresponding page with all inputs of the jBPM business process is illustrated in Figure 41.

If the user starts the risk evaluation the process manager SB will invoke the above mentioned method in the SLSB for the risk evaluation. The risk event is calculated by the given parameter and after the successful run, the result value is persisted in the `RiskEvent` and presented at the value evaluation page as shown in Figure 42.

Adding a new Risk Event - Value Evaluation

Description	Value
Risk Event	
-> id	5
-> description	Malfunction
-> limit value	0.2
-> probability RE	0.1826585726304366
-> risk horizon	1.0
-> start date	2010-01-15
Risk Category	
-> id	1
-> description	Malfunction
-> limit type	Reliability
Economic Resource	
-> id	19
-> description	TA 2500
Risk Control	
-> id	4
-> description	induce a maintenance
Risk Model	
-> id	4
-> description	Reliability model
Value Entity	
-> id	7
-> description	Average life-time
-> value	6.0
-> value UoM	ALT

Result: Reliability=0.8173414273695634

Enter Abort

Figure 42: The presented result of the jBPM process

To finalise the jBPM business process, the user has to press the “Enter” button, then all entities will be persisted, the jBPM business process determines and the risk event is added in the risk event overview table, which is illustrated in Figure 36.

4.3. Integrated ERM – Loss Given Risk Event

When assessing financial instruments, the maximal loss is a reliable financial ratio to estimate the inherent risk. In the following Chapter the implementation details for evaluating the LGE is presented.

4.3.1. Entity Beans: specifying the Data Structure

The general database structure for ERM is defined in Chapter 4.1.1. The required parameters for the LGE computation and their corresponding EB are specified as follows:

- The economic resource
  - the according financial instrument, for which the maximal loss shall be estimated
  - is connected with the `RiskEvent` through a composition
- The limit
  - the critical value which causes the alert if the maximal loss is greater than this boundary
  - is recorded as a property of the `RiskEvent`
- The probability of risk event
  - denotes the estimated probability at which the risk event will take place
  - is recorded as a property of the `RiskEvent`
- The volatility
  - defines the volatility of the risk factor interest increase in percent
  - is persisted as a `ValueEntity` of the `RiskModel`
- The confidence coefficient
  - specifies the space of the negative trend of the underlying risk factor
  - is persisted as a `ValueEntity` of the `RiskModel`
- The risk control
  - denotes the action, which shall be taken in the case of exceeding the limit
  - is linked to the `RiskEvent` through a compositon

The values for the `RiskModel`s are persisted by an own EB, the `ValueEntity`. This approach enables to record as many values as necessary for a single `RiskModel`. On the other side, the `RiskEvent` has a predefined number of properties. Additional properties cannot be extended, but in this case it should not be required because the most common are already implemented.

4.3.2. Session Beans: implementing the Business Logic

To evaluate the LGE according to the explanations in Chapter 3.2, the SLSB `RiskEvaluationBean` provides two methods analogous to the reliability model.

Code 17: Evaluation methods for the duration model

```
/**
 * Method for evaluating the maximal loss of a given RiskEvent.
 * All necessary entities are fetched from the DB and checked.
 * The actual values are computed, if possible.
 * Otherwise the prior computed values are taken to compute the result.
 * All computed values are persisted in the corresponding EBs.
 * Persists the LGRE in the RiskEvent entity.

 * @param riskEvent the RiskEvent with the necessary values
 * @return the LGRE (the maximal loss of the given RiskEvent)
 * @throws Exception in case of wrong/missing values
 * @see RiskEvent
 * @see RiskModelEntity
 * @see ValueEntity
 */
public double evaluateLGRE(RiskEvent riskEvent) throws Exception {
     ... }

/**
 * Method for evaluating the maximal loss of a given FI.
 * detailed description:
 * {@link RiskEvaluationBean#evaluateLGRE(RiskEvent)}

 * @param financialInstrument the EconomicResource must be an instance
     of FinancialInstrument with the associated RiskEvent
 * @return the LGRE (the maximal loss of the given FI)
 * @throws Exception in case of wrong/missing values
 */
public double evaluateLGRE(EconomicResource financialInstrument) throws
Exception { ... }
```

Code 17 illustrates an extract of the SLSB `RiskEvaluationBean`, which shows the commended methods for the LGE evaluation. The difference is again the parameter in the `evaluateLGRE` method, which can be an instance of `FinancialInstrument` or of `RiskEvent`. The method stores all calculation parameters in the corresponding EBs, computes the result and returns it to the invoking process manager SB.

4.3.3. jBPM: orchestrating the Business Processes

jBPM Business Process and Pageflow

The jBPM business process and the according pageflow for both implemented risk models are explained in Chapter 4.1.3.

Process Manager

As described in Code 16, the process manager SB has also a unique procedure for both risk models. The limit type differentiates both risk models, when the result of the actual risk event shall be computed.

4.3.4. JSF: presenting the Graphical User Interface

In Chapter 4.1.4 similar steps in the jBPM business process are illustrated till the second task-node.

To continue the process steps, task-node number three asks for the risk event properties as depicted in Figure 43. The first is the limit, which shall not be exceeded, in monetary units. The second value is the probability of the risk event, which describes the probability of occurrence.

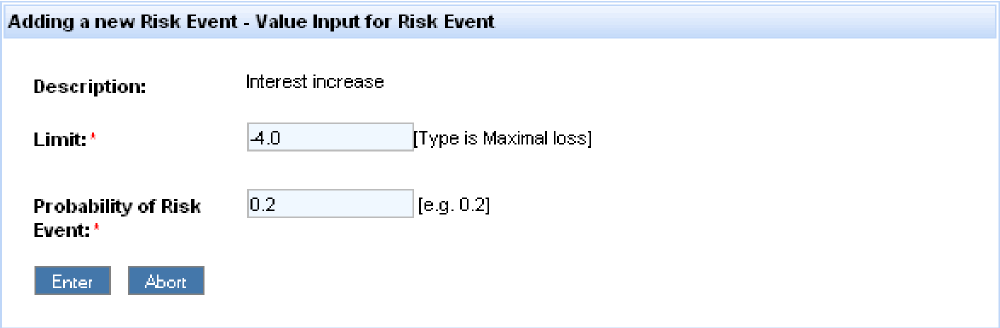


Figure 43: Value input page for the LGE evaluating risk event

In the fourth process step the user has to enter the parameter for the duration model. To estimate the risk factor, the volatility and the confidence coefficient are required. The other risk model parameter like the future cash flows, nominal rate, etc. are fetched from the database as explained in the evaluation process in Chapter 3.2.2.

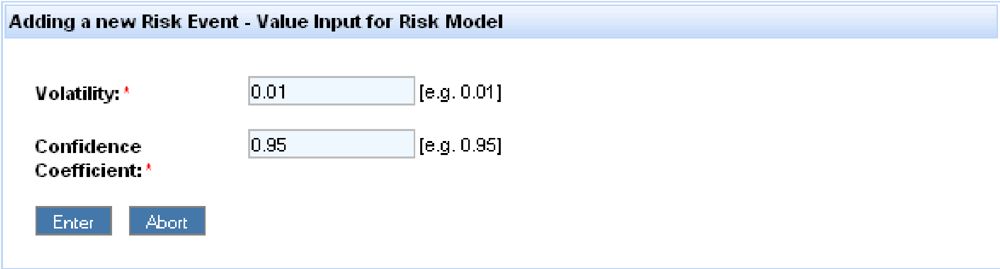


Figure 44: Value input page for the duration model

The next task-node asks the user for the risk control, which specifies the action, that will be executed, if the limit is exceeded.

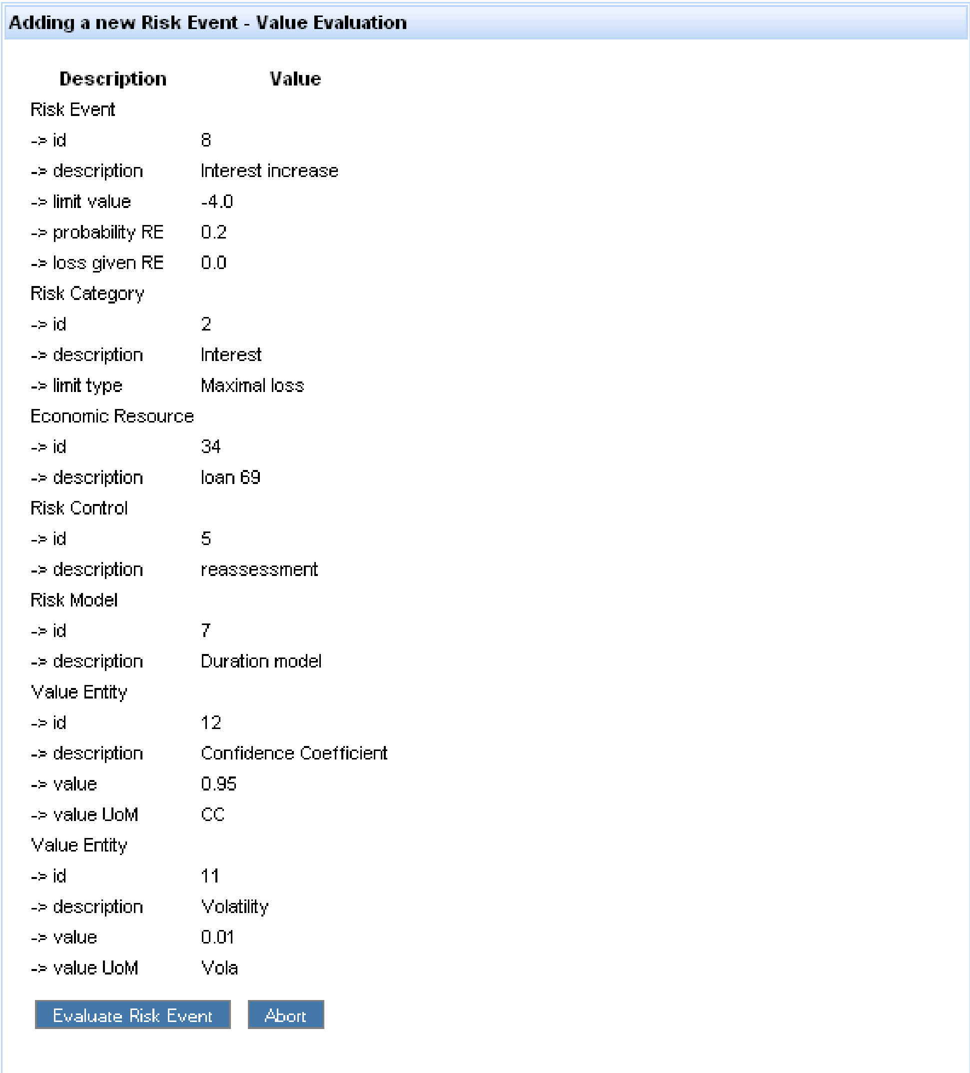
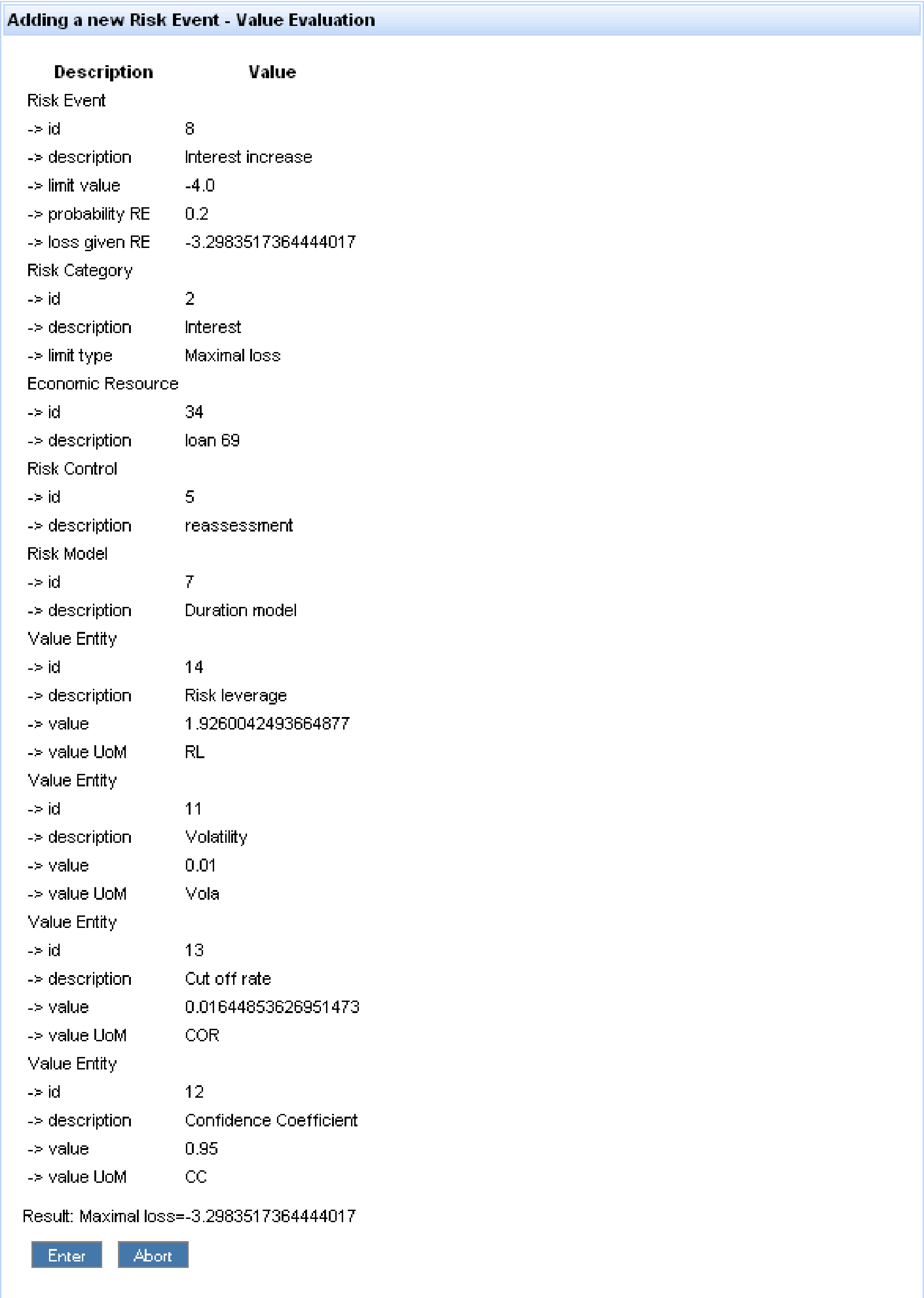


Figure 45: Last check of the input values before the risk evaluation

In the sixth and last process step the risk evaluation will be computed. But first the input values have to be checked, a summary of all input values of this jBPM business process can be seen in Figure 45.



Adding a new Risk Event - Value Evaluation

Description	Value
Risk Event	
-> id	8
-> description	Interest increase
-> limit value	-4.0
-> probability RE	0.2
-> loss given RE	-3.2983517364444017
Risk Category	
-> id	2
-> description	Interest
-> limit type	Maximal loss
Economic Resource	
-> id	34
-> description	loan 69
Risk Control	
-> id	5
-> description	reassessment
Risk Model	
-> id	7
-> description	Duration model
Value Entity	
-> id	14
-> description	Risk leverage
-> value	1.9260042493664877
-> value UoM	RL
Value Entity	
-> id	11
-> description	Volatility
-> value	0.01
-> value UoM	Vola
Value Entity	
-> id	13
-> description	Cut off rate
-> value	0.01644853626951473
-> value UoM	COR
Value Entity	
-> id	12
-> description	Confidence Coefficient
-> value	0.95
-> value UoM	CC

Result: Maximal loss=-3.2983517364444017

Enter Abort

Figure 46: Final page for the LGE evaluation

The last page for the LGE evaluation is illustrated in Figure 46. In the last line the result with the calculated LGE is shown. The attentive viewer of the screenshot might have noticed that the `ValueEntity`s have been expanded. The cut off rate and the risk leverage have been added to the existing confidence coefficient and the volatility. The reason is that these parameters are necessary for the evaluation of the LGE, therefore they are stored as additional `ValueEntity`s in the database.

After this final task-node, in which the result was evaluated and the parameters were persisted, the jBPM business process will forward to the end-state and determine. The new created risk event can be seen together with the existing risk event in Figure 47.

Risk Event - Management

Back to overview
Add new Risk Event

Id	Description	Closed/Locked	Limit Type	Start Date	Risk Horizon	Probability	Loss	Related Resource	[Description]	
1	Malfunction	☐	0.5	21.01.11	1.0	15.4 %	0.0	Equipment	[TA 2500]	Edit disable
2	Interest increase	☐	-4.0		0.0	20.0 %	-3.2983517364444	BulletLoan	[loan 69]	Edit disable

Figure 47: Overview of all created risk events in ERPControl

4.4. Integrated ERM – Risk Reporting

After the risks were assessed by the evaluation of the risk events, the results have to be monitored. The monitoring is done by comparing the risk event results with their limits.

Risk Event - Report

Id	Description	Limit Type	Limit Value	Probability	Loss	Related Resource	[Description]	Status
1	Malfunction	Reliability	50.0 %	84.6 %		Equipment	[TA 2500]	●
2	Interest increase	Maximal loss	-4.0		-3.30	BulletLoan	[loan 69]	●
9	Malfunction	Reliability	25.0 %	23.9 %		Equipment	[ZA 100]	●
10	Interest increase	Maximal loss	-4.0		-4.66	BulletLoan	[loan 69]	●

Figure 48: Risk report in ERPControl

If a risk event of type “Reliability” is assessed, the limit will be the boundary, which shall not be under-run. Analogous if a risk event has the limit type “Maximal loss”, the computed LGE shall not exceed this maximum in a negative sense.

Figure 48 presents four risk events, two of each risk model. The first two risk events were just created in the above explained jBPM business processes and both are within their defined limits, therefore the status of these lines is green. The last two risk events were additionally added to illustrate the functionality of the risk reporting tool and these risk events exceed their limits, which leads to a red status ball for these lines.

If the risk manager creates new risk events or reevaluate the existing entities, the status of each, active risk event will be displayed immediately. This simplifies the risk management and saves more time for estimating the risk model parameters or corrective measurements to avoid financial damage and to increase the value of the enterprise.

5. Conclusion

Enterprise risk management is a powerful tool to monitor existing risks in an organisation. Its usage is essential for every size of enterprise, but to benefit from it, a comprehensive combination of assessment, evaluation, measurement and monitoring of the various types of risks is required. COSO established an integrated framework, which specifies the tasks and assists the implementation of ERM in enterprises.

But knowing, which risks may occur, and having a guideline, what shall be done, is only one side of the coin. The challenge is to automate ERM, so that a risk manager can create the risk events, define the risk models and give the parameters that the ERP system can execute the monitoring. If an observed risk takes a turn for the worse, the ERP system will notify the risk manager from it, who can work against the negative trend before something unpleasant happens.

ERPControl is a next generation, REA-based management information system, which has the ability to become an essential benchmark for future ERP systems. The integration of reports for ERM in ERPControl, which was presented in this thesis, expands the functionality of this scientific software project and can be the basis of future developments.

Two different risk models have been implemented: the first is the reliability model that observes the probability of a malfunction of the underlying device. The probability of failure is evaluated by using the exponential distribution, whereby the actual reliability is compared to a given limit. As time goes by, the present reliability has to be reevaluated because of the ageing of the device. Through this measurement the monitored equipment can be maintained or replaced before a failure occurs.

The second implemented risk model deals with financial instruments and their maximal loss because of interest changes. The interest rate is the focused risk factor, because it can cause financial loss in the future. To evaluate the maximal loss, the discounted future cash flow is assessed by two financial ratios corresponding to a generic risk model. The resulting maximal loss can be taken to estimate the risk potential of the underlying financial instrument.

Due to the generic approach of the integration of reports for ERM in ERPControl related work such as the extension of the existing risk models is enabled and shall be arranged through future projects. Based on the enhancements of the various contributions to this ambitious ERP system, future developments will be implemented much faster, because of the already integrated functionality, the supported business processes and the finished projects.

6. References

1. **Gorry, G. Anthony; Scott Morton, Michael S.** A Framework for Management Information Systems. s.l. : Sloan Management Review, 1971, Vol. SMR Classic Reprint, pp. 49-61.

2. **McCarthy, William E.** The REA Accounting Model: A Generalized Framework for Accounting Systems in a Ahared Data Envirionment. s.l. : The Accounting Review, 1982, Vols. LVII, No. 3, pp. 554-578.

3. **ISO/IEC JTC 1/SC 32Nnnnn.** *ISO/IEC 15944-4:2006 Information technology - Business Operational View -- Part 4: Business Transaction scenarios - Accounting and economic ontology.* 2006.

4. **McCarthy, William E.** *REA Enterprise Ontology Source Page.* [Online] 07 September 2002. [Cited: 13 Jänner 2011.] https://www.msu.edu/user/mccarth4/rea-ontology/.

5. **Hruby, Pavel.** *Model-Driven Design Using Business Patterns.* s.l. : Springer, 2006.

6. **Kobialka, M.; Miazhynskaia, T.; Rodler, Ch.; Schwaiger, W.** *Business Performance Management-Information System: Enterprise Control Architecture and Web-Implementation.* s.l. : Vienna University of Technology - Institute of Management Sciences - Corporate Finance, Planning and Control, 2010.

7. **ISA-International Society of Automation.** *Enterprise-Control System Integration Part 1: Models and Terminology.* 2000. ANSI/ISA-95.00.01-2000.

8. **Commitee of Sponsoring Organizations.** *COSO Web Portal.* [Online] [Cited: 15 01 2011.] http://www.coso.org/.

9. **Commitee of Sponsoring Organizations of the Treadway Commission.** *Enterprise Risk Management - Integrated Framework.* s.l. : www.cpa2biz.com, September 2004.

10. **Müller, Bernd.** *JBoss Seam - die Web-Beans-Implementierung.* s.l. : Hanser, 2008.

11. **Bauer, Christian; King, Garvin.** *Java Persistence with Hibernate.* s.l. : Hanser, 2007.

12. **Farley, Jim.** *Practical JBoss Seam Projects.* s.l. : Apress, 2007.

13. **Modarres, M.** *What Every Engineer Should Know About Reliability and Risk Analysis.* New Yourk : Marcel Dekker, Inc., 1993.

14. **Schwaiger, Walter S.A.** *Controlling unter Unsicherheit.* s.l. : Institut für Managementwissenschaften - Fakultät für Maschinenwesen und Betriebswissenschaften - TU Wien, 2010.

15. **Schwaiger, Walter S.A.** *Grundlagen der Betriebswirtschaft.* s.l. : Institut für Managementwissenschaften - Fakultät für Maschinenwesen und Betriebswissenschaften - TU Wien, 2010.

16. **Achleitner, Stefan.** *Web 2.0 based ERP System for Planning and Control of Financial Instruments.* 2010.

17. **Fellner, Daniel.** *Modellbasierte Planung und Steuerung unter Unsicherheit.* 2010.