Universität des Saarlandes - Philosophische Fakultät II

Semantically-enriched Business Process Modeling and Management

Dissertation
zur Erlangung des Grades eines
Doktors der Philosophie
der Philosophischen Fakultäten
der Universität des Saarlandes

vorgelegt von
Alexandra Weissgerber
aus Forbach, Frankreich

Saarbrücken, 2011

Der Dekan
Herr Univ.-Prof. Dr. Erich Steiner

Berichterstatter
Herr Univ.-Prof. Dr. Jörg Schütz
Herr Univ.-Prof. Dr. Johann Haller

Tag der letzten Prüfungsleistung: 20 Juli 2011

Acknowledgements

To my sons, Mathieu, Philippe and Victorien

Writing a dissertation is a long and stony road. In my opinion, it cannot be achieved without the support of beloved people. So my first thanks go to my family. My sons, who had to cope with my full time plan and my partner Patrick Ziegler who assumed kindly many of the daily tasks. Beside of his cooking talents, I really appreciated our professional exchanges where he shared his expert knowledge about enterprise management with me. I also thank my parents, Monique and Jean-Marie Weissgerber who gave me the possibilities to absolve my computer science study and supported me with their proudness, a feeling which was, shared by my grandma Elvire Zéwé and my sister Gabrielle Weissgerber. My partner's parents, Hiltrud and Hubert Ziegler supported me with the most valuable resource that I needed: time. They often took care of my little baby that will be 15 months when I will submit this dissertation. I also thank Sylvain Maik who offered me a MacBook, which was a really valuable programming computer.

This dissertation is the result of experiences gathered through my professional life. Friends, colleagues and professional relations that I have met all these years have contributed to the result in a significant way. After my computer science study – and here I do not want to forget my professors at the IUT in Metz who personally got very involved in their students success –, I worked 10 years at the research institute IAI in Saarbrücken. During all this time, I have felt like being member of a big family. I thank particularly the whole IAI team that enabled me to acquire all the linguistic competence that I needed to absolve my master in computational linguistic (DESS Industries de la langue). The IAI was managed jointly by Prof Dr Jörg Schütz and Prof Dr Johann Haller. Prof Dr Haller gave me the chance to start at the institute even though I came to the interview accompanied by my mother! I am very happy that he accepted to be my second thesis supervisor. He supported me with feedback and information. I especially want to mention Dieter Maas who was a pillar of this institute and my colleague and friend Véronique Hinz. I also will never forget Christoph Horschmann who has departed from us far too soon.

It was a very hard and heartbreaking decision to close the chapter IAI. As I decided to follow new challenges, I had the luck to enter another great company with great colleagues, the IDS-Scheer AG at Saarbrücken, which focuses on Business Process Management and Modeling. I was part of the development team and later development manager of the ARIS SOA Architect product (shortly described in this dissertation). The sensibility for language quality control I have acquired during my IAI years made me quickly recognize the problems that occur if no quality control is performed during modeling tasks, independently of the modeling environment. It is how the idea of this dissertation was born. I had several discussions with colleagues who gave me trust in the benefits of my work. Even though the dissertation was not financed and performed jointly with the IDS-Scheer, some of my colleagues provided me with training material and reference data that enabled me to test my approach.

Sometimes we meet people who have a positive influence on our objectives in an unexpected way. I met Gerhard L. Düwer in a soft skill seminar (called 'OFFBOX 1') that deals with leadership and self-management. He gave me a simple, but useful advice, which I want to share with the readers of my dissertation. I have started my thesis with the development of a prototype that gave me confidence in the necessity and feasibility of my approach. At this point, I did not really write something and suffered from the white page syndrome. I could not figure out how to write scientifically. Then Gerhard said to me: 'Do not try to write with a high quality. Write your ideas in your own language. Write only three sentences per day. If you can't, write two or even one.' I followed this advice and wrote three lines per day during one week in telegraphic style followed by a description of what I had implemented. This was the beginning of my dissertation.

The final touch was given by Carola Niklaus, my proofreader, who contributed to the correction on language level and gave me according advice for future works. I warmly thank her!

Last but not least, my first and foremost thanks are devoted to my doctoral thesis supervisor, Prof. Dr. Jörg Schütz. He supported me since my first steps in this direction, when I still thought that I am not able to do a thesis. He gave me feedback that enabled a continuous improvement of my work, encouraged me to write papers and made me discover conferences. Even if in the first moment I did not appreciate some comments, I realized that he was right after according improvements. 'Jörg, your Champagne cave is now full... Let us start another one...'

Contents

Nomenclature

API .... Application Programming Interface

BAM .... Business Activity Monitoring

BI .... Business Intelligence

BPEL .... Business Process Execution Language

BPEL4WS .... Business Process Execution Language for Web Services

BPM .... Business Process Management

BPM .... Business Process Modeling

BPMI .... Business Process Management Initiative

BPMN .... Business Process Modeling Notation

CIM .... Computationally Independant Model

CMS .... Content Management System

CORBA .... Common Object Request Broker Architecture

CRM .... Customer Relationship Management

CWE .... Collaborative Working Environment

DAML+OIL ..... Darpa Agent Markup Language + Ontology Inference Layer

DCOM .... Distributed Component Object Model

DMS .... Document Management System

DoDAF .... Departmentof Defense Architecture Framework

EA .... Enterprise Architecture

EAF .... Enterprise Architecture Framework

EAI .... Enterprise Architecture Integration

EMF .... Eclipse Modeling Framework

EMS .... Environment Management System

EPC .... Event-Driven Process Chain

ESB .... Enterprise Service Bus

eTOM .... enhanced Telecom Operations Map

GGG .... Global Giant Graph

HTTP .... HyperText Transfer Protocol

IaaS .... Infrastructure As aService

IDE .... Integrated Development Environment

ISO .... International Standards Organization

IT .... Information Technology

ITIL .... Information Technology Infrastructure Library

KB .... Kknowledge Base

KM .... Kknowledge Matrix

KPI .... Key Performance Indicators

LDAP .... Lightweight Directory Access Protocol

LE .... Linguistic Engine

LSP .... Localization Service Provider

MDA .... Model-Driven Architecture

MoDAF .... Ministry of Defence Architecture Framework

MOF .... Meta-Object Facility

N3 .... Notation3

NP .... Noun Phrase

OASIS .... Organization for the Advancement of Structured Information Standards

oAW .... openArchitectureWare

OAXAL .... Open Architecture for XML Authoring and Localization

OMG .... Object Management Group

OWL .... Web Ontology Language

PaaS .... Platform As aService

PIM .... Platform Independant Model

PLM .... Product-Lifecycle Management

PSM .... Platform Specific Model

QA .... Quality Assurance

QM .... Quality Management

QMS .... Quality Management System

RDF .... Resource Description Framework

RDFS .... Resource Description Framework Schema

RIF .... Rule Interchange Format

SaaS .... Software As aService

SCC .... Supply-Chain Council

SCM .... Supply-Chain Management

SCOR .... Supply-Chain Operation References

SOA .... Service Oriented Architecture

SOA-RM .... Service Oriented Architecture - Reference Model

SOAP .... Simple Object Access Protocol

SPARQL .... SPARQL Protocol and RDF Query Language

SQL .... Structured Query Language

SWRL .... Semantic Web Rule Language

TOGAF .... The Open Group Architecture Framework

TQM .... Total Quality Management

TURTLE .... TTerse RDF Triple Language

UML .... Unified Modeling Language

URL .... Uniform Resource Locator

W3C .... World Wide Web Consortium

WfMC .... Workflow Management Coalition

WPDL .... Workflow Process Definition Language

WS-BPEL .... Web Services Business Process Execution Language

WSDL .... Web Services Description Language

WSML .... Web Service Modeling Language

WSMO .... Web Service Modeling Ontology

WWW .... World Wide Web

XLIFF .... XML Localization Interchange File Format

XMI .... XML Metadata Interchange

XPDL .... XML Process Definition Language

Kurzfassung

Um heute am Markt erfolgreich zu agieren, müssen alle an diesem Markt operierenden Unternehmen und - insbesondere die Keyplayer - sich den Gesetzen von Flexibilität und Effektivität unterordnen. Diese Unterordnung ist absolut notwendig, um dem schnellen Wechsel der Marktanforderungen an Unternehmen und ihren Produkten sowie den neuen ökonomischen Herausforderungen folgen zu können.

Die Methoden des Geschäftsprozessmanagement und ihre Grundlage, die Geschäftsprozessmodellierung, haben sich mittlerweile als Werkzeuge zur adäquaten, effektiven und effizienten Analyse durchgesetzt. Mit ihrer Hilfe werden das Management der Geschäftsprozesse und deren operative Umsetzung, die Interoperabilität, der translinguale und transkulturelle Austausch und insbesondere der wichtige Part der Kommunikation zwischen den an Projekten beteiligten Mitarbeitern bewältigt. Die Beschreibung des eigenen Handelns, die Abbildung der Unternehmensabläufe mittels Geschäftsprozessmodellierung, ist heute der Schlüssel zum Erfolg.

Die Geschäftsprozessmodellierung ist eine Methode um intuitiv die Organisationsstrukturen und die abhängigen Geschäftsprozesse mit Hilfe von Modellen zu beschreiben. Die Modelle bestehen aus Geschäftsobjekten und werden in einer dedizierten Datenbank abgelegt, dem Business Repository. Geschäftsprozessmodelierungswerkzeuge mit statischen Modellen wie Organisationsstrukturen, IT-Infrastrukturen, Systemlandschaft und Datenmodellebene erlauben die Definition von dynamischen Ablaufmodellen in einer standardisierten Notation – entweder BPMN von OASIS oder EPK. Die Beschreibungen liefern die Basis für das Geschäftsprozessmanagement, das aus zyklisch wiederkehrenden Phasen wie agile Analyse, Monitoring sowie (Neu)-Design und Optimierung von Geschäftsprozessen besteht.

In der Vergangenheit unterstützte eine BPM-Software die manuelle Feststellung von Engpässen und Optimierungspotential in den betrachteten Geschäftsprozessen. Heute, in Zeiten zunehmender Prozessautomatisierung und automatisierter Workflow Engines in distribuierten Umgebungen wie z.B. Service Orientierten Architekturen (SOA) oder Cloud Computing Szenarien, werden Prozessanalyse, Monitoring und Optimierung auch zunehmend automatisiert.

Die ultimative Vision von modernem Geschäftsprozessmanagement ist die sofortige Umsetzung (Produktivsetzung) neuer Prozessschritte, sobald sie auf der Geschäftsseite modelliert und von der Unternehmensführung durch eine virtuelle Prozessanalytik

abgenommen wurden.

Falls die bestehende IT-Infrastruktur nicht über die passenden Services (Funktionalitäten und Softwarekomponenten) verfügt, wird eine reaktive Implementierung der vorhandenen Module oder die Beschaffung (build or buy) neuer Funktionalitäten erwartet, die genau den optimierten Geschäftsanforderungen entsprechen.

Notwendigerweise sollten alle diese Aktivitäten in einem passenden Rahmen platziert werden, zum Beispiel einer Workflow Engine, um eine flexible Ablaufsteuerung und vor allem ein effektives Monitoring zu ermöglichen. In der Realität laufen die Prozesse innerhalb einer Workflow Engine nur nach intensivem manuellem Aufwand, da die Komplexität der Arbeit zur Verknüpfung von technischen und betriebswirtschaftlichen Parametern – Ergebnisse von Modellierung und Optimierung – eine enge Zusammenarbeit von Personen aus den unterschiedlichsten Wissensbereichen des betroffenen Unternehmens erfordert.

Zu Zwecken der Dokumentation und Kommunikation werden zusätzlich Teile des Geschäftsprozessmodelles im Intranet des Unternehmens publiziert. Diese Veröffentlichung führt zu offenkundigen - oftmals erstmals erkennbaren - sprachübergreifenden und multikulturellen Aufgabenstellungen, weil in global operierenden Unternehmen ein großer Teil der Geschäftsobjekte in natürlicher Sprache benutzt wird.

Die neuen Aufgabenstellungen sind die Prüfungen aller vorliegenden sprachanhängigen Arbeitsergebnisse von BPM auf Sprachkompetenz und auf Geschäftsperformanz. Sprachkompetenz subsumiert syntaktische und semantische Korrektheit, terminologische Konsistenz, Lokalisierung und Übersetzungslesbarkeit. Geschäftsperformanz inkludiert Geschäftsangemessenheit, indem natürlichsprachliche Ausdrücke eindeutig Geschäftsobjekte benennen und die Geschäftssemantik im Hinblick auf logische Beziehungen zwischen verschiedenen Geschäftsartefakten geprüft wird. Die Nichteinhaltung der Sprachkompetenz und Geschäftsperformanz von Objekten des BPM führt zu Inkonsistenzen und Redundanzen im Business Repository. Dies verhindert eine direkte und nahtlose Transformation in IT-Lösungen, eine brauchbare Lokalisierung und insbesondere das Verschmelzen und Aggregieren schon lokalisierter Geschäftsmodelle - eine der wichtigsten Voraussetzungen für eine erfolgreiche weltweite Expansion.

Da der gesamte Geschäftsprozesslebenszyklus durch solche Defizite offensichtlich stark beeinträchtigt wird, muss die Rahmenorganisation des Geschäftsprozessmanagement, die Governance-Ebene, unbedingt um Modellierungsrichtlinien auf Inhaltsebene für die Objektbenennungsaufgaben erweitert werden.

Das Ziel ist es, ein Qualitätssicherungsmodell und eine passende Metrik für die Benennung der Geschäftsobjekte zu etablieren und neue Methoden zu entwickeln, die die komplette Einhaltung der gewonnenen Metrik gewährleisten.

Parallel dazu existiert die zwingende Anforderung die Lücke zwischen Geschäftsprozessen und IT-Prozessen umgehend zu schließen sowie die Interoperabilität der Teilnehmer – Menschen und Maschinen – am globalen BPM zu maximieren.

Die vorliegende Dissertation enthält innovative Lösungen zu bisher nicht vollständig behandelten Aufgabenstellungen wie der Verifizierung der Spracheebenen sowie einem besseren Alignment zwischen Modellierungsebenen und Umsetzungsebenen (IT Lösungen). Sie lässt eine Wissensmatrix entstehen, die dem klassischen BPM eine automatisierte, maschinenverarbeitbare neue Schicht – die semantische Ebene – oberhalb des Business Repository hinzufügt.

Natürliche Sprachverarbeitung und statistische Merkmalsmodelle unterstützen den Aufbau dieser Wissensmatrix mit Hilfe der Formalismen und Fähigkeiten des Semantic Web. Um die Vollständigkeit, Genauigkeit und Beschränktheit der Inhalte der Wissensmatrix zu sichern, wird die notwendige Zusammenarbeit der am BPM beteiligten Analysten mittels spezialisierter, inhaltbezogener Kollaborationsplattformen ergänzt und mit maschinell gestützten Lernmethoden unterstützt.

Alle Phasen des BPM-Lebenszyklus werden unterstützt und es entsteht zusätzlich ein Qualitätssicherungsmodell, um vorliegende und neuentstandene Regeln (durch Analyse vorliegender Referenzmodelle) aus Sicht der Governance-Anforderungen effektiv einzusetzen. Die Einhaltung dieser Regeln wird durch eine automatisierte Umgebung gewährleistet, die Bestandteil der neuentstandenen Lösung ist. Diese Unterstützung der Konsistenz- und Richtlinienanforderungen trägt zu einer marginalen Verbesserung des Aligments aller Projektteilnehmer bei. Ebensowichtig für die Sicherstellung eines effektiven Aligments ist das zukunftsorientierte Design der Arbeit, welches eine Integration der neuentstandenen Dienste natürlich auch in Service Orientierten Architekturen ermöglicht.

Die innerhalb der Dissertation entwickelten Methoden und Tools unterstützen auf herausragende Weise die bisher in BPM-Projekten fast immer vermisste Durchgängigkeit (Hauptanliegen der Unternehmen) der von Forschung und Beratungsdienste angebotenen Lösungansätze.

Abstract

In rapidly changing business environments, new economic and market challenges must be analyzed, managed and set into operation in an appropriate, effective and efficient way by global industrial players. Flexibility, effectiveness and effectivity are keys to acting as a successful leader in global markets.

One of the major methods used to continuously adapt the processes and workflows of an enterprise to new business environments and versatile business requirements is Business Process Modeling and Business Process Management – both abbreviated BPM. BPM helps to manage business processes and their operational implementation as well as interoperability, translingual and transcultural interchange and communication between the employees participating to the project. The description of the own handling with Business Process Modeling is the key to success.

Business Process Modeling is a means to intuitively describe the enterprise's organizational structure and the interrelated business processes with the help of diagrams that are composed of Business Objects and that are stored in a dedicated database – the Business Object Repository. The BPM software with static models of an enterprise's organizational chart, IT infrastructure, system and process landscape, and business entities allows the definition of dynamic workflow models in a standardized notation – either BPMN of OASIS or EPC – , which constitutes the basis for Business Process Management, which consists in periodical phases like agile analysis, monitoring, (re)design and optimization of the business processes.

In the past, BPM software supported the manual detection of bottlenecks and optimization potentials in the envisioned business processes. Nowadays, since processes are more and more automated and runnable in dedicated workflow engines within distributed environments such as for example Service Oriented Architectures (SOA) and cloud-enabled computing landscapes, the process analysis, monitoring and optimization tasks are automated, too.

The ultimate vision of modern process management is that as soon as processes are modeled on the business side and approved by management through virtual process analytics they can be set into operation and executed immediately. If the IT infrastructure does not consist of appropriate components and facilities, a reactive implementation or procurement of capabilities (build or buy) that meet exactly the business requirements is expected. In reality, however, process models run in workflow

engines only after intensive extra manual efforts because it is necessary to include much additional technical information business process analysts are normally not aware of.

In addition, parts of the Business Process Model are published in an enterprise's intranet for documentation purposes to serve different human process participants. This multi-purpose situation even leads to cross-language and cross-cultural aspects within globally operating enterprises because a large set of the Business Objects that occur in process models is labeled with natural language expressions.

These object labels are neither checked in terms of their language competence nor in terms of their business performance. Language competence subsumes syntactic and semantic correctness, terminological consistency, and localization and translation readiness. Business performance includes business appropriateness, which means the natural language expressions unambiguously label the Business Objects, and business semantics in terms of logical relationships between the different business artifacts. For the Business Repository, the dissatisfaction of language competence and of business performance of the object labels leads to inconsistencies and redundancies, and prevents, on the one hand, a direct and seamless transformation into IT solutions, and on the other hand, a usable localization and translation into other cultural settings for foreign enterprise branches, subsidiaries and partners, as well as the merging and aggregation with already localized business models – one of the most important conditions for a world-wide expansion.

Since the whole enterprise process life cycle is affected by these shortcomings, it is recommended to extend the Business Process Management Governance with modeling guidelines for the content level of the labeling task. The objective is to develop a quality assurance model and a metric that apply to the Business Object labeling, and methods to ensure the full compliance of the metric thereby also bridging the gap between business processes and IT processes and fostering the interoperability between global process participants – man and machine.

This dissertation describes innovative solutions which cover the topics that are not exhaustively handled like the verification on language level as well as a better alignment between modeling levels and implementation levels (IT solution). The solution creates and employs a Knowledge Matrix that adds a machine processable semantic layer on top of the Business Object Repository.

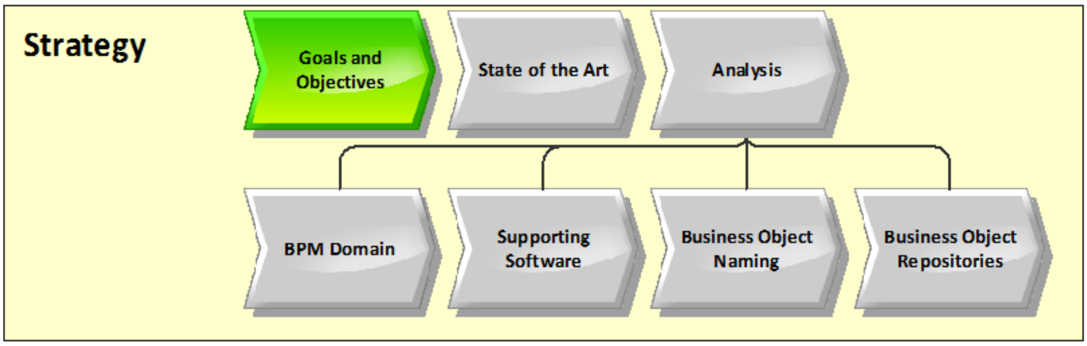
Natural language processing and statistical feature models contribute to the building of the matrix by using standard Semantic Web formalisms and capabilities. To ensure exhaustiveness, accuracy as well as restrictiveness of the Knowledge Matrix content, business analysts additionally cooperate through specialized content collaboration platforms that are supported by machine learning techniques.

All BPM life cycle phases are addressed. Additionally a quality assurance model is developed from a governance requirements point of view. This model is composed of existing rules and newly developed ones issued from the analysis of available reference models. The guidelines compliance is ensured by an automated environment that is part of the new solution. The support for consistency and guidelines account for a better alignment of all process participants.

The developed components are designed to be deployed as a dedicated service in a Service Oriented Architecture. The methods and tools implemented in this dissertation support in an outstanding manner the continuity and alignment that are missing in the solutions proposed by consulting and research approaches.

Chapter 1

Introduction



Business Process Management is an area which gains very high interest. In most companies, the active setup within enterprises has only started recently, in particular combined with the introduction of Service Oriented Architecture (SOA) competencies. Appearance and content of Business Process Management activities are multifaceted regarding their establishing, development and actual deployment, but the overall objective remains the optimization of enterprise business processes to gain on productivity and, in turn, reduce production and management costs.

Business Process Modeling is a key method to achieve enterprise business process performance. Modeling business processes in dedicated models makes assessments about the enterprise structure and business operations: organizational structure, IT infrastructure, system landscape and the relations between the business entities. In this context, modeling means representing the knowledge gained about the organization with business objects which are specialized objects that feature a specific semantics. The main purpose of such models is information dissemination within the organization, documentation of processes for quality management certifications and detecting optimization potentials to achieve a continuous process improvement.

Business objects are created in business models provided by a modeling environment and are stored in a business repository. The so-created models and business repository are the fundaments of Business Process Management related tasks. The business process analyst names the business objects with string literals which should reflect the purpose of the object as close as possible. The free naming possibilities are not restricted by modeling environments. The hypothesis of this dissertation is that free business object naming leads to inconsistencies in the business repository and affects the whole Business Process Management life cycle. Recent studies emphasized the importance of the quality of labels contained in business process models. Jan Mendling and Hajo Reijers [1] explored labeling styles in the SAP Reference model (English language) to prove following assessment:

Yet the semantic and pragmatic quality of process models is hardly investigated. In this context, the choice of an appropriate text label has presumably a significant impact on the pragmatic quality of a model.

Nicolas Peters and Matthias Weidlich [2] denote the importance of the understandability of process models for human interactions:

Conceptual models are mostly used for human to human communication. Besides several other aspects, that is, the chosen modeling notation or the model layout, the labeling has a strong influence on the understandability and, therefore, the quality of a process model. Consequently, labels should be reused and aligned across different process models, whereas similar labels such as homonyms should be avoided.

Henrik Leopold et al. [3] highlight that business process modeling involves people having different technical backgrounds and who may not be strong-skilled in modeling activities:

Many companies have recently expanded their business process modeling projects such that often thousands of process models are designed and maintained. The problem with such initiatives is that staff with limited experience takes part in modeling, with a detrimental effect on the quality of the models. The quality of activity labels, i.e., the textual descriptions in a process model, is an important quality aspect in this context.

All these studies concentrate on the readiness and understandability aspects in human-to-human interactions. As business process models are an important communication mean within organizations, this dissertation addresses human interactions as well, but is not limited to this topic. The impact of the quality of business object labels on the whole Business Process Management life cycle is analyzed, including the automation and execution of business processes in a SOA.

The objective is to create a quality management framework which offers a quality assurance model (set of metrics), tools, methods, governance processes and best practices for ensuring the quality of business objects denominations. The framework development complies with the Business Process Management life cycle. The figure 1.1 outlines the approach. Parts of this model will be included preliminarily to the sections which deal with the according topic. The 'Strategy' phase encompasses the goals definition and different topics of analysis that prove the needs for such a framework

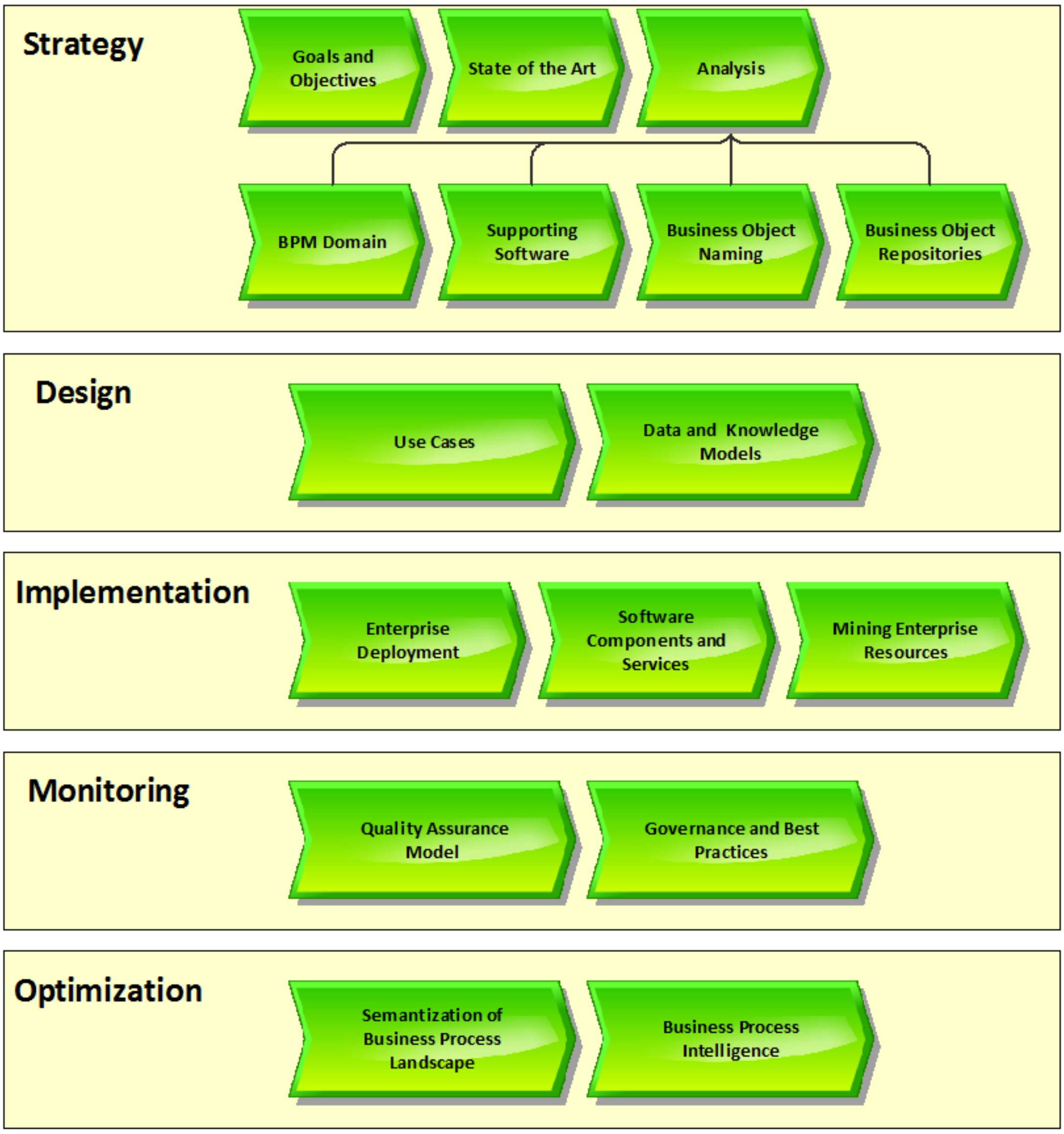


Figure 1.1: Development Stages of the Quality Management Framework

and determine the technologies necessary for the design and the implementation:

- *Goals and Objectives*: the overall objectives are defined in this introduction.
- *State of the Art*: standards are very important for the global success in this area. The chapter 3 describes the state of the art of the application domain. The first section exposes current trends of enterprise architectures, which have evolved to distributed computing like enterprise service bus, Service Oriented Architecture or cloud-computing landscapes as well as crowdsourcing aspects (Web 2.0) that increasingly support daily business tasks. The second section introduces business process modeling formalisms standards or de-facto standards, norms for quality management and the most common process management frameworks. The solution is based on semantic technologies which are part of

the semantic web. The last section describes relevant semantic web standards and recommendations.

- *Analysis – BPM Domain* : the chapter 2 gives a short introduction of the Business Process Management domain to give the readers a first preview of the application scenario.

- *Analysis – Supporting Software*: the chapter 4 describes all the software used to support this dissertation, for programming the prototype as well as the use of a modeling environment and knowledge representation software.

- *Analysis – Business Object Naming*: the chapter 5 provides an analysis of the language related problems in the modeling process and the impacts to the Business Process Management activities in each stage of process management life cycle including the process governance level. The area of the dissertation considers both monolingual and translingual aspects.

- *Analysis – Business Object Repository*: the first part of the chapter 7 analyses the content of real-life business repositories. Statistics empowered by linguistic intelligence are performed on the stored business objects in order to extract valuable content and examine the diversity, complexity and range of naming problems.

The 'Design' phase includes the process of establishing design requirements from a user point of view and the research and conceptualization process for designing the underlying knowledge models:

- *Use Cases*: the chapter 6 describes the necessary roles for human resources and the general architecture of the solution as well as system requirements from a user and governance point of view for each stage of the process management life cycle. The chapter includes a detailed design of graphical user interfaces enriched with enterprise relevant examples that are demonstrating a panel of difficulties inherent to natural languages.

- *Data and Knowledge Models*: the analysis of business repositories serves as a foundation for building business-specific knowledge models with help of machine learning techniques. The second part of the chapter 7 describes the data structure of the knowledge models which are used as a back-end by the implemented software service.

The 'Implementation' phase describes the software components included in the quality management framework and their deployment in order to fulfill the requirements of the use cases as described in the chapter 6.

- *Enterprise Deployment*: the first part of the chapter 8 describes the deployment of the implemented software components as a service and its interactions within the Business Process Management application scenario.

- *Software Components and Services*: the second part of the chapter 8 describes the software components that ensure the compliance of the business object denomination with the metrics defined in the quality assurance model. The developed service is composed of software annotators combining linguistic intelligence with semantic technologies to extend the existing business repositories with a machine understandable semantic layer. The annotators support different reasoning capabilities to gain implicit knowledge from the business-specific knowledge models.

- *Mining Enterprise Resources*: enterprise structured or unstructured resources support the modeling of business processes, give detailed information about specific process steps or have to be considered on the process management governance level. The chapter 9 explains how knowledge included in process documentation, corporate terminologies and data warehouses can enrich the process management life cycle and extend the business-specific knowledge models. The developed techniques are adapted for interpreting natural language service descriptions included in Web services to favor service discovery in a Service Oriented Architecture.

The 'Monitoring' phase defines methods and a set of metrics used for controlling the quality of the business object denominations.

- *Quality Assurance Model*: the first part of the chapter 10 summarizes the guidelines and metrics gathered from information material and deduced from the analysis of the business repository objects. A quality assurance model is derived from the facts and conclusions issued from the previous chapters.

- *Governance and Best Practices*: the second part of the chapter 10 describes governance processes and best practices for introducing and integrating semantically-enriched BPM applications in enterprise modeling strategies.

The 'Optimization' phase assesses the fulfillment of the objectives defined in the 'Strategy' phase and seeks for new optimization potential.

- *Semantization of Business Process Landscape*: the first part of the conclusion summarizes the results achieved so far by enriching the Business Process Management life cycle with semantic techniques.

- *Business Process Intelligence*: objective of the quality management framework is to achieve a continual improvement of the quality assurance model and related tools and processes. The second part of the conclusion introduces optimization potential like the context-related aspects of a process model, business intelligence facets and adaptation strategies of the monolingual solution into global business scenarios in order to motivate future challenges and development work.

Recent research addresses some of the aspects which are deepened in this dissertation. Many of the cited studies have performed or tested their approach on the SAP

Reference Model, which consists in a collection of process models, which comprises about 20000 activities in the English language. Jan Mendling and Hajo Reijers [1] discuss some issues of labeling using activity samples of the SAP reference model. They classified the labels in three categories:

1. Verb-Object style, which represent the main part of the activity labels. The verb denotes the action and the noun the object on which the action is performed. Ambiguities can occur when the same word can both be a noun and a verb ('Process Cost Planning').

2. Action-Noun Style, where the action verb is used as a noun ('Notification Printing').

3. Rest category, which neither includes a verb nor a noun that refers to an action.

Based on this classification, the authors wanted to verify the two following hypotheses:

H1: Verb-object style labels are less frequently perceived as being ambiguous, followed by action-noun style labels, and finally rest labels.
H2: Verb-object style labels are perceived as the most useful by process model readers, followed by action-noun style labels, and finally rest labels.

They selected a process model from the practice containing all cited types of labels and conceived a survey asking students of a process modeling course about their perceptions of ambiguity and usefulness of certain labels contained in the model. The survey confirmed both hypotheses, suggesting that the verb-object style should be used as a general guideline for modeling business processes and that a deviation from the verb-object rule is likely to decrease the clarity of the model.

Henrik Leopold et al. [4] have performed a study on the English activity labels from the SAP Reference Model. They have investigated how various natural language processing (part-of-speech tagger) and ontologies like WordNet can be used to detect activity labeling styles in an automatic fashion. They used two kinds of metrics:

- Based on the length of activity labels: labels that are too long decrease the readability. Labels containing only one word or no word referring to an action will most likely cause misunderstandings. The length distribution shows that most labels have length of only three words and that there is no label having more than 12 words. The average label length equals to 3,78.

- Based on the frequency of activity labeling styles: in a previous study, Jan Mendling et al. [5] have performed a manual analysis of the SAP Reference Model. About 60% of labels follow the verb-object style, 34% were classified as action-noun labels and only about 6% of the labels belong to the rest category. This study serves as a reference for evaluating the techniques developed by Leopold et al. Their results show that high precision and recall can be achieved automatically by using part-of-speech tagging techniques. WordNet, used to assign grammatical information to words, even leads to better results.

This study and the underlying one demonstrate the needs of a quality management framework for business object denominations. It shows that even reference models are conceived with many ambiguities.

In an other study, Henrik Leopold et al. [3] proposed a method for automatic refactoring of English action-noun activities into labels of verb-object, signifying the same action performed on the same business object. The method was evalutated on English labels issued from the SAP Reference Model. Their approach is based on three steps:

1. Label style recognition: a script recognizes a set of label properties, like the usage of the gerund form or prepositions (for example 'of') and categorizes the label into particular styles.
2. Derivation of an action and a business object from the label: WordNet is used to learn whether the first word is a verb and which infinitive it has.
3. Composing a verb-noun label: simple step of concatenating the infinitive verb and nouns.

This approach is an algorithm-based method which does not involve natural language processing tools. Only some predefined label structures are concerned. After evaluation of the previous steps, the refactoring was performed with a precision of 85%.

Fabian Friedrich [6] focuses on the understandability of the labels. He studied how the lexical database WordNet can help in identifying process model labels that could lead to misunderstandings. The approach is applied on the SAP Reference Model. The previously cited papers concentrated on the label structure, while this study focuses on the term usage.

WordNet contains 155 000 words organized into synonym sets. Specific relations like synonyms, homonyms, hypernyms/troponyms, meronyms ('part-of' relationship) and antonyms can be retrieved in an automatic way. The first problem addressed in the study is the use of several words with the same meaning, as this contradicts the principle of a shared vocabulary and increases the ambiguity of labels. A stemming algorithm is used to unify different word forms and enable the synonym acquisition through WordNet. The second addressed problem arises when different people use different levels of abstraction to describe the business object, detected via the hypernym relations in WordNet.

The study highlighted many label inconsistencies. Anyway, the approach is limited:

Words in WordNet can have many different meanings and the usage of averages can lead to unwanted results. On the one hand, this can distort the consistency quality as a synonym might just not be appropriate for the word under investigation. On the other hand, it could lead to a depth for the specificity metric which deviates strongly

from the one the specific meaning intended for this word has. Thus, it is necessary to determine the meaning of a word within a label prior to calculating its quality.
...
some domain specific terms that are completely understandable for a domain expert are not contained in WordNet.

Nicolas Peters and Matthias Weidlich [2] have already described in 2008 an approach for generating automatically a glossary from reference models (evaluated in particular with the SAP Reference Model) to ensure a consistent usage of labels. They assume that reference models contain consistent and precise models with labels of high quality. They compared two different glossary generation approaches: a fine-granular glossary containing data objects related to actions and a glossary that contains complete phrases that are directly applied as labels for process model elements. The first approach would allow an easier glossary management and an effective control of the labeling activities. However, to overcome the automatic speech tagging issues (to identify verbs from nouns), they chose the second approach. A label is approved if it is part of the glossary, which, in turn, was generated from the reference model. The glossary entries are indexed so that labels that contain a certain semantic closeness can be proposed if the used one is not allowed.

In subsequent works [7], the process of generating a glossary is extended by taking dependencies in terms of co-occurences of labels into account and considering the control flow characteristics of the process models from which the glossary is created. The underlying assumption is that labels typically follow some kind of implicit ordering like 'Receive invoice' that will typically occur before 'Archive invoice'. The advantage is that the automatically generated glossary provides support in terms of label suggestion features that consider structural and control flow aspects. The control flow aspects will not be relevant in this dissertation.

The authors describe the two limitations of their approach:

Clearly, a more fine-granular approach that builds upon single terms and some definition of valid relations between them (e.g., phrase structures) can be assumed to be more effective. Various of the labels that are not contained in our glossary directly might be derived by combining single terms of different labels that are already part of the glossary.

Our approach focuses on a consistent usage of labels rather than on consistency between labels. We ensure that labels are applied correctly with respect to the type of the respective model element as well as co-occurrence and control flow dependencies between them. Still, our approach does not offer any control on the actual creation of labels and the consistency between different labels. Consequently, flaws such as misspellings, incomplete labels, labels that contain control flow information, or the usage of synonyms are not addressed by our approach. Hence, it is of particular importance that the labels from which the glossary is generated are of high-quality.

In this dissertation, the conceptual choices made for the development of the quality management framework are exactly the opposite. One of the objectives is to build data and knowledge models based on single business terms with defined relations between them to ensure a consistent usage of labels as well as a consistency between labels. The knowledge models will then be extended by a knowledge matrix containing complete business object labels in order to reduce computing time (simple lookup mechanism). These entries are created based to the restricted entries of the single term glossaries, ensuring that every occurring label meets the defined quality metrics.

Patrick Delfmann et al. [8] described an approach that has some similar fundamental aspects to this dissertation. Their article presents the results of a label analysis issued from 4805 process models of a concrete German modeling project. Before modeling, the involved business analysts were provided with naming conventions and glossaries. The results showed that existing conventions that assure the process model readability and comparison are insufficient. Six causes for label conflicts were identified:

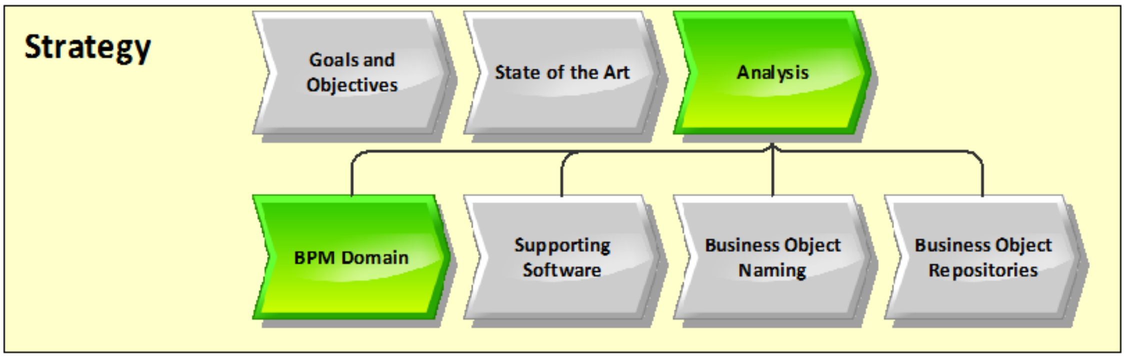
- Usage of synonyms including an inconsistent usage of abbreviations
- Usage of meta-information in business objects that refer to the process model structure, like 'Process end'
- Usage of labels that do not fit the element type (events instead of activities)
- Inconsistent usage of sentence structures
- Usage of aggregated labels: two different activities or events that are joined in one label (for instance usage of 'and')
- Common spelling errors

Their solution is based on a domain-specific glossary that contains nouns, verbs, adjectives and adverbs which are used in process models. Additionally, a set of accepted sentence structures are defined. A technique which assures the conformity of the input with the convention is provided by replacing the sentence structure elements by glossary entries. Free-text input is still allowed. A specific linguistical parser transforms the input in parts, like sentence structure and base forms (lexical units) of the used terms. The parts are checked against the domain-specific glossary. If not available, potential synonyms are searched in general glossaries and checked again against the domain-specific glossary. The valid terms are proposed to the modelers.

This approach is similar to the core method applied in this dissertation concerning the usage of a linguistical component and glossaries. The cited article highlights the required efforts to establish such a domain-specific glossary and defines this fact as the main limitation of the approach. The business repository analysis performed in this dissertation (German language) will help to derive a set of common business terms that can be reused in different modeling projects. Governance processes which support the extension of the terminological resources in a collaborative way will be defined. Concrete modeling guidelines will be derived from the analysis results. None of the cited studies focused on the whole process life cycle and the impact on process automation. This dissertation will close this gap as well.

Chapter 2

The Domain of Enterprise Business Management



Today, the business environment is characterized by highly concurrent and rapidly changing markets. Companies have to respond with their products and services faster than their competitors do. They also face increasing pressure to cut costs and raise their productivity, without neglecting the quality. The business management techniques cannot rely only on the only financial and accounting indicators, like benefit, cash flow, stock management... These economic indicators are results and do not allow to detect early enough the structural problems and risks which can affect the company's health. Also, financial security laws like the Sarbanes-Oxley Act in USA or its equivalents in other countries (LSF, Loi de sécurité financière in France) lead to better and faster risk identification to keep the trust of the customers, shareholders, suppliers or banks.
Delivering products and services of high quality to customers becomes a main strategy for companies. Certifying the product conformity against norms like the ISO 9000 family represents a strong selling argument and becomes even more and more a customer requirement. To fulfill the quality issues, some companies introduced quality control procedures like six sigma, originally developed by Motorola.

All companies, regardless of their size, produce products and services in a defined manner. Under the constraints defined in the previous paragraphs, companies need methods and technologies to analyze the way they are working. Two established management models are widely spread out in companies and get increasing importance: Business intelligence (BI) and Business Process Management (BPM).

2.1 Business Intelligence

Companies need to analyze changing trends in markets, customer's preferences or behavior to develop innovative products or services in anticipation of customer's changing demands. Also, knowledge of companies' internal metrics like sales amount and productivity issues helps the management to take accurate decisions. The term 'Business Intelligence' refers to applications and technologies used to gather, store and analyze data and information about companies operations. BI technologies must be able to deal with huge amounts of data.

The first step consists in collecting the data. Sales and production metrics are often available in a structured form (table entry of a database, specialized software repositories...), while information on customers' trends or satisfaction have to be retrieved from miscellaneous data sources. Therefore, BI software is conceived to gather large quantities of unstructured data. BI software often relies on the use of Key Performance Indicators (KPI) . KPIs are business-related metrics used to quantify objectives and to control the performance of an organization in a previously defined strategy. Each organization has to identify which KPIs reveal strategic assessment for their business; therefore KPIs differ from one company to another. The main criteria for determining KPIs are summed up by the acronym SMART:

- Specific
- Measurable
- Achievable
- Result-oriented
- Time-based

This means that a specific business process must be defined with clearly defined objectives. At given periods, the KPIs values must be available to estimate whether the planned results are achieved or whether they are on the right way for achievement. If not, weaknesses must be analyzed and decisions for adjustment should be taken.

The next step after information gathering is information storage. If the amount of data is too huge to be stored in one database, several distributed data warehouses are employed. Data warehouses consist of a collection of several databases or databases servers, often distributed by different providers. The data warehouse encompasses access management for different users or user roles. For instance, marketing managers may have access to different information than accounting managers.

BI systems analyze the stored data in order to generate reports or statistic information, which should improve the business decision making. While quantitative KPIs (for example financial metrics) are quite simple to get and to evaluate, non-quantitative metrics represent an interesting information material for companies. Assuming that BI systems are able to extract the needed information from appropriate sources, it remains difficult to transform it in quantitative metrics for evaluation. A typical example illustrating this problem is to measure the staff satisfaction within its organization.

The BI reports and information are available in a defined time-span. However, companies increasingly depend on always up-to-date information. BI 2.0 is a term representing the future of BI. Its purpose is to close the time gap in order to provide real-time data. Monitoring KPIs in real-time is called 'Business Activity Monitoring' (BAM).

2.2 Business Process Management (BPM)

Johannson et al. [9] give following definition of a business process:

A set of linked activities that takes an input and transform it to create an output. Ideally, the transformation that occurs in the process should add value to the input and create an output that is more useful and effective to the recipient either upstream or downstream.

All companies have business processes, regardless of size or type. Three types of business processes can be distinguished:

- *Management processes*: consist of processes that govern the operations of a system, including strategic decision-making by the top management.
- *Operational processes*: these processes are repetitive tasks performed by an organization which are part of core business. They create the primary value stream of the company. In a pharmaceutical company, one operational process represents the steps performed from the research to the commercialization of the medicament.
- *Supporting processes*: these processes support the operational processes. Typical examples are the human resources department, or the accounting tasks.

The medium and long-term success of a company is driven to a significant extent by the quality of its process management. BPM is a management model and an approach that allows organizations to manage and continuously improve their business processes.

BPM covers following fields:

- *Process design*: encompasses the capture and documentation of existing processes and the design of the to-be processes. This step should ensure that a correct and efficient design is prepared. The design phase answers the questions 'Who does what, in what sequences, what services are provided and which software systems are used in the process?'.

- *Process modeling*: creation of the flow and interactions in specific visual formalisms of the repository objects created in the design phase.

- *Process execution*: consists in developing and deploying operations in order to automate the process as much as possible. Not all steps are automatization candidates: some tasks must further be executed by persons. These tasks are called human tasks.

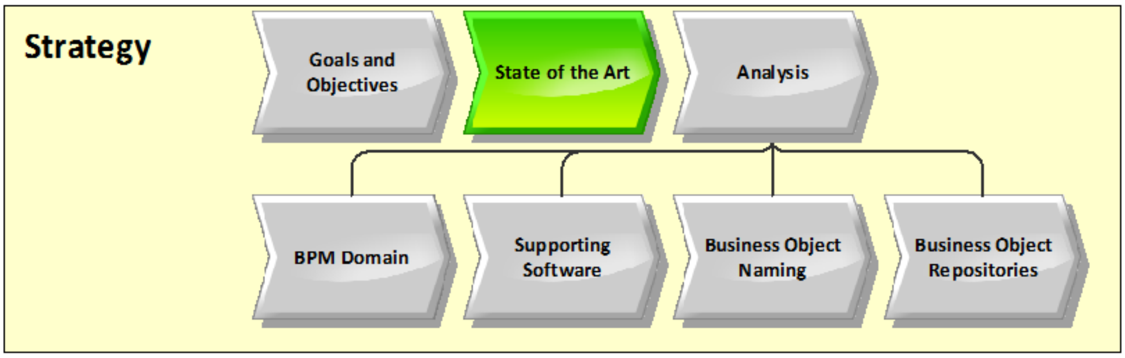
- *Process monitoring and controlling*: encompass tracking and determining the state of the process or values used by different process steps. This information feeds statistical procedures, which permits the evaluation of the process performance.

- *Process optimization* is based on the performance information provided by the monitoring step and on the knowledge gathered during the modeling stage. This knowledge helps to identify the current bottlenecks and give indications on potential optimization activities in order to save time and costs.

Process optimization closes the process management loop, called '*Process life cycle*'. The term loop or life cycle is employed because the optimization phase often implies to revise the process design.

Chapter 3

State of the Art



Business Process Management encompasses a broad spectrum of disciplines, standards or enterprise-wide initiatives. Depending on the organization's strategy, multiple ways can be taken to achieve business process performance and the alignment of the IT processes to business objectives. This chapter describes the state of the art concerning business-relevant topics and techniques employed to implement the solution detailed in this dissertation:

- Overall description of enterprise architectures and related topics.
- Business Process Modeling standards including formalisms and reference models.
- Introduction to Quality Management relevant standards.
- Collaborative work and more generally Web 2.0 techniques which are increasingly integrated in the design of enterprise processes.
- Semantic web techniques which are employed to add a semantic layer over the modeled processes to give them a signification.

3.1 Enterprise Architectures

3.1.1 Enterprise Architecture Fundamentals

The specialists of information technology (IT) are increasingly confronted with the requirements of customers and the challenges to explain the use and the benefits of B2B, EAI, BPM, SOA... to them. The need of an overall frame to host and to combine these technologies as center of an architectural methodology rapidly leads to installing Enterprise Architecture (EA) as the umbrella of all upcoming IT integration projects. The broad spectrum of EA fields is illustrated by the article '10 Definitions of Enterprise Architecture. Which corresponds to Yours' [10] that gathers ten different EA definitions from experts.

The EA related tasks consist of a description of the organization's structure and dynamics including business goals, business processes, organizational structures, software applications and IT infrastructure as well as the relationship between all the assessed business entities and artifacts. Enterprise architects use various analytics, monitoring techniques and conceptual tools to analyze and to document the organization's architecture. EA related tasks are supported by standard enterprise architecture frameworks (EAF), which gather methods, tools, process reference models used by architects to describe the architecture of an organization. The Open Group Architecture Framework (TOGAF) [11], one of the most established EAF, is based on four pillars: business architecture, applications architecture, data architecture and technical architecture. Domain-specific EAFs are developed to fit organization's special needs like for example the Department of Defense Architecture Framework (DoDAF) [12] customized for the US defense department and for defense industries or the UK Ministry of Defence Architecture Framework (MoDAF) [13], the british counterpart of DoDAF.

All EAFs agree that the business strategy acts as an input for IT governance so that the IT infrastructure, software implementation and application procurement really fit the business needs. Jeanne W. Ross et al. [14] have analyzed why business strategies and IT capabilities are difficult to align:

First, the strategy isn't always clear enough to act upon. General statements about the importance of 'leveraging synergies' or 'getting close to the customer' are difficult to implement. So the company builds IT solutions rather than IT capabilities. Second, even if the strategy is clear enough to act upon, the company implements it in a piecemeal, sequential process. Each strategic initiative results in a separate IT solution, each implemented on a different technology. Third, because IT is always reacting to the latest strategic initiative, IT is always a bottleneck. IT never becomes an asset shaping future strategic opportunities.

The second remark explains why heterogeneous IT systems and IT infrastructure landscapes are growing in organizations. Enterprise Application Integration (EAI) is a method and architecture principle to integrate the disparate software components into a homogeneous system landscape. EAI recommends the use of middleware products that foster the communication and data exchange between the different applications and favor the seamless execution of business processes. The newer generation of EAI called Enterprise Service Bus (ESB) is an architecture that exploits Web Services and other XML-based messaging standards. Unlike EAI, the ESB proposes a completely distributed integration with help of service containers.

ESB has gained more relevancy with the emergence of cloud-computing. Cloud-computing is a concept which allows the externalization of computer infrastructure (Infrastructure as a Service – IaaS) of platforms including databases or runtime systems (Platform as a Service – PaaS) and of applications (Software as a Service – SaaS). Even if cloud-computing discharges organizations from many IT management costs and expensive license fees, the heterogeneous IT landscape is now extended to external providers.

Another aspect is the integration of the Internet of Things, that means that intelligent and physical objects can communicate among themselves and with computer systems over internet technologies. The objectives of the 'ADiWa' [15] research project financed by the German 'Bundesministerium für Bildung und Forschung' is to link events issued from the real world with BPM, in order to create, manage or adapt complex business processes.

The next two subsections describe the main concepts used as a strategy to facilitate EAI and to bridge the gap between business goals and IT landscape.

3.1.2 Service Oriented Architecture

3.1.2.1 SOA Concepts

The Service Oriented Architecture (SOA) is a concept, a style for designing architectures using services. The Organization for the Advancement of Structured Information Standards (OASIS) SOA Reference Model (SOA-RM) [16] defines SOA as:

a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.

The SOA-RM specification bases its definition of SOA around the concept of 'needs and capabilities', where SOA provides a mechanism for matching needs of service consumers with capabilities provided by service providers.

The SOA concept is based on services. Standard dictionaries are defining a service as:

1. *An act of helpful activity; help; aid: to do someone a service.*

2. *The supplying or supplier of utilities or commodities, as water, electricity, or gas, required or demanded by the public.*

The OASIS SOA-RM [16] defines service as:

a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description.

An important concept emerging from this definition is that services are providing capabilities. A capability is defined by standard dictionaries as:

1. *A talent or ability that has potential for development or use.*
2. *The capacity to be used, treated, or developed for a specific purpose: nuclear capability.*

The OASIS defines capability as:

A resource that may be used by a service provider to achieve a real world effect on behalf of a service consumer.

Consumers have specific needs. The OASIS SOA-RM [16] defines need as:

A need is a measurable requirement that a service participant is actively seeking to satisfy. The aspects of a need are that it can be measured and that it belongs to a participant.

If the needs (also called requirements) are matching the capabilities of the service providers, the consumers could use the service if they are authorized to.

Services in a SOA are reusable and interoperable components, like modules or components in a standard modularization concept. One difference with modules is that services are loosely coupled. Loosely coupled means that services have no hard dependencies to each other, like a linker, which links modules written in a programming language. Services are interoperable via exchange formats, often via an XML format.

Services are not dependent of any technologies or development platform. They may be implemented with technologies like Web Services, Common Object Request Broker Architecture (CORBA) [17], Distributed Component Object Model (DCOM) [18], but may not be implement at all. In this case, the service may be a specific task, executed by an organization unit or a person. It is also called 'manual task'.

The SOA concept can be considered from different point of views: from the business point of view and the IT point of view.

From the business perspective, business analysts want to define the strategy on the business side to increase flexibility. Flexibility is ensured if every process step is coupled with a service. In the ideal case, existing services are reused if a process is redesigned so that no additional efforts remain at the IT side. If a new process step is created, the business analyst can model a service object on the process step and define the service requirements. This serves as a service blueprint, which is used by the IT development team to develop or find appropriate services.

From the IT perspective, SOA means the reuse or redesign of existing functionality (modules, components) as a service to favor interoperability. A formalism like BPEL (see section 3.2.5) allows services to be orchestrated.

The challenge is to bridge the gap between the business view and the IT point of view so that both levels are totally aligned. Needs and requirements are designed at the business side, the capabilities (in this case services) are created or found on the IT side. The SOA manifesto [19] emphasizes that business value is a priority of SOA and summarizes a few guiding principles service orientation should follow.

3.1.2.2 Web Services Description Language

The Web Services Description Language (WSDL) is a XML formalism used to describe Web Services. The current version of WSDL is 2.0 and was formerly called 1.2. The WSDL specification comprises three parts, the primer [20], the core language specification [21] and the specification of predefined extensions [22].

As an Application Programming Interface (API) delivered with a software, the WSDL provides an interface definition for Web Services operations with their input and output parameters, the parameter types and the thrown exceptions. The implementation is managed by the service provider and is not part of the WSDL definition. The service provider exposes only the interfaces and the address (endpoint) where the Web Services can be called, typically a Uniform Resource Locator (URL) [23] or HyperText Transfer Protocol (HTTP) [24] string.

The binding section of the WSDL file defines how the Web Service can be accessed using the underlying Simple Object Access Protocol (SOAP). In the current version 1.2 of the specification [25], the acronym has been dropped. SOAP defines a protocol for exchanging messages in the XML format over the network.

3.1.2.3 Web Service Modeling Ontology

The Web Service Modeling Ontology (WSMO) [26] provides a conceptual framework and a formal language for semantically describing all relevant aspects of Web services in order to facilitate the automation of discovering, combining and invoking electronic services over the Web. It supports research efforts towards so-called Semantic Web services, that is, enriching Web services with machine-processable semantics.

The WSMO describes four aspects: the goals (objectives when consulting a Web Service), ontologies, mediators (interoperability between ontologies) and Web Services under the aspect of semantic description. Example ontologies can be found on the main web page of the WSMO working group [27].

The Web Service Modeling Language (WSML) is a formal language that provides a syntax and semantics for the WSMO. The WSML specification [28] defines the goals of WSML as follows:

The WSML aims at providing means to formally describe all the elements defined in WSMO. The different variants of WSML correspond with different levels of logical expressiveness and the use of different languages paradigms.

3.1.3 Model-Driven Architecture

The Model-driven architecture (MDA) is a software design approach for software development launched by the Object Management Group (OMG). MDA is based on the definition of platform-independent models which serve as an input to transformation processes in order to create executable code. The OMG executive summary [29] highlights that MDA addresses challenges like platform portability, cross-platform interoperability and platform independence, and in turn reduces the costs and time for development and improve the application quality.

The OMG MDA guide [30] details the MDA concepts. To ensure a platform-independent design, a Computation Independent Model (CIM) is created from a computation independent viewpoint, which focuses on the system environment and system requirements. The CIM is amended by a Platform Independent Model (PIM), which focuses on the operations of a system while hiding the details necessary for a particular platform. A platform independent view shows the part of the complete specification that does not change from one platform to another. The Platform Specific Model (PSM) combines the definition in the PIM with the details which specify how that system uses a particular type of platform.

The three types of models are transformed to provide a platform-specific software. Many methods can be used for transformation. Some approaches are transforming a CIM into a PIM and a PIM into a PSM by interpreting and including platform-specific attributes, which are modeled as well in an ideal case. Other approaches are directly transforming a CIM or a PIM into deployable code by including for example the platform-specific attributes in the transformation code templates (transformation patterns).

A platform independent view may use a general purpose modeling language or a domain-specific language which fits to the area in which the system will be used. To achieve that objective, the OMG created the Unified Modeling Language (UML) whose currently adopted version is 2.3. The UML 2.0 Infrastructure specification [31] defines the foundational language constructs required for UML 2.0. It is complemented by the UML 2.3 Superstructure [32], which defines the user level constructs required for UML 2.3. The UML 2.3 comprises 14 diagram types; seven represent structural information and the other seven represent behavior and interactions.

To define and specify the UML language, the UMG has developed a metamodeling architecture, the Meta-Object Facility (MOF) [33]. MOF is an architecture including four layers:

- M0, meta-metamodel layer: defines the language in which UML is defined. MOF can be used for defining other metamodels. This layer is also called 'MOF-layer'.
- M1, metamodel layer: layer where UML and other kind of metamodels are defined according to the M0 specification.
- M2, model layer: layer where users are defining their own models with the metamodels defined in M1. Typical examples are UML models like UML class diagrams included in an IT concept.
- M3, system layer: layer where the runtime instances are defined, for example concrete instances of an UML class.

MOF is supported by an XML-based exchange format called XML Metadata Interchange (XMI) used for exchanging metamodels on the M0-, M1-, or M2-Layer. XMI has become an international standard described by the ISO/IEC 19503:2005 norm [34].

MDA conceived in particular for software engineering purposes is nowadays increasingly used to cover business needs and thus bridge the gap between business analysts and software engineers. Michael Guttman and John Parodi [35] have demonstrated the links between MDA and SOA with six case studies. Their main idea is to capture the business logic models in high-level business models and realize the process elements as service-oriented modules with help of MDA. An analog approach is detailed in the section 3.2.

3.1.4 Additional Enterprise Management Considerations: Enterprise 2.0

3.1.4.1 Enterprise 2.0 Fundamentals

The term 'Enterprise 2.0' denotes the use of 'Web 2.0' technologies within an organization in order to support business processes. The term 'Web 2.0' encompasses technologies which enhance the classical World Wide Web features with social-networking capabilities, which enable collaboration and interoperability between internet actors, humans, machines or software agents.

In the position paper 'What is Web 2.0' [36], Tim O'Reilly describes the Web 2.0

pillars and gives following lesson:

Network effects from user contributions are the key to market dominance in the Web 2.0 era.

In the web 2.0 summit (2009), Tim O'Reilly and John Battelle [37] still emphasize the role of collective intelligence and the systems which support collective intelligence:

Web 2.0 is all about harnessing collective intelligence.

The Enterprise 2.0 binds Web 2.0 technologies into intranet and extranet portals which link employees, customers and suppliers in a social network to achieve the business goals in a more efficient way by favoring collaboration and information sharing. Andrew P. McAfee [38] first introduces the term 'Enterprise 2.0' in an article published in the 'MITSloan Management Review' magazine:

These new digital platforms for generating, sharing and refining information are already popular on the Internet, where they're collectively labeled 'Web 2.0' technologies. I use the term 'Enterprise 2.0' to focus only on those platforms that companies can buy or build in order to make visible the practices and outputs of their knowledge workers.

In his book, Andrew P. McAfee [39] analyses the business value of Enterprise 2.0 techniques and gives recommendations to introduce them in companies. He highlights following benefits:

Enterprise 2.0 offers significant improvements, not just incremental ones, in areas such as generating, capturing, and sharing knowledge; letting people find helpful colleagues; tapping into new sources of innovation and expertise; and harnessing the 'wisdom of crowds'.

The wisdom of crowds is defined as the phenomenon where the aggregation of information issued from a group is often more valuable than the information issued from a single person, even an expert. Collaboration is the key prerequisite to give an added value to Enterprise 2.0 technologies. Organizations can then outsource tasks by making an open call to an undefined group of people or a community, called the crowd. This strategy is called 'crowdsourcing'. The best example remains the free encyclopedia 'Wikipedia'. The term 'crowdsourcing' was first introduced by Jeff Howe [40] in the 'Wired' magazine:

Technological advances in everything from product design software to digital video cameras are breaking down the cost barriers that once separated amateurs from professionals. Hobbyists, part-timers, and dabblers suddenly have a market for their efforts, as smart companies in industries as disparate as pharmaceuticals and television discover ways to tap the latent talent of the crowd. The labor isn't always free, but it costs a lot less than paying traditional employees. It's not outsourcing; it's *crowdsourcing.*

All these definitions and expected benefits demonstrate the necessity for organizations to set up collaborative working environments and integrate them in their business processes in order to keep a leading position in the market.

3.1.4.2 Collaborative Working Environments

A collaborative working environment (CWE) supports members of an organization and related process participants in their individual and cooperative work. Web communication technologies favor a successful outsourcing strategy and the cooperation between geographically dispersed teams (virtual teams). The main CWE components are communication and crowdsourcing applications.

Besides traditional e-mails, instant messaging, calendaring, contact sharing and videoconferencing applications, newer communication methods have emerged, like twitter or blogs. Twitter [41] is a short messaging service (microblog) which allows to send small messages of up to 140 characters called tweets. The Twitter Inc. provides the Twitter platform accessible from the Twitter home page or other channels like mobile phones. Besides sending messages, people can subscribe to categories of interest to receive real-time news. Michael Rosemann [42] has posted an article in the ARIS community which describes how Business Process Management can benefit from Twitter, in particular under the process change management aspect. Although companies may not want to be dependent on external services like Twitter for security reasons, they may adopt internal microblogging middleware.

A blog (from web log) is a part of a website organized by theme where blog participants create entries sorted by date in the ascending or descending order. Each item can contain text, multimedia elements and links to further resources. Private people often use blogs to post anecdotes about their life, whereas groups or communities are sharing points of view about topics of interest. In the business context, blogs are means to share opinions about enterprise relevant themes, exchange tips and tricks for solving daily minor problems or to involve customer's opinions in a product development process. The blog keeps by nature traces of the conversation so that each topic can be consulted by persons encountering a similar problem or who are interested in that topic.

A wiki is a part of a website which allows the editing of collaborative documents. Authorized users can add, modify or delete parts of documents. In the editing mode, wiki comes with a light markup language which defines the document layout. When the reviewer considers the document as completed, a lock mechanism prevents from further editing. Enterprises wikis support several business processes, like sales support, project support, informal learning, product planning or customer service.

The previously described synchronous and asynchronous collaboration applications are standard features of collaboration platforms. Collaboration platforms can be seen as a portal where the communication applications are integrated into portlets (container for displaying web content) and accessible to users according to their access rights. They include the functionalities of Document Management Systems (DMS) and Content Management Systems (CMS) to facilitate content sharing. Some collaboration platforms are featuring workflow engines which support the execution of business tasks. Product or service providers are more and more understanding that collaboration platforms can promote their offers. For example in the business process management domain, the ARIS community [43] and ARISalign [44] are inviting users to share process models, experiences and points of view about different BPM topics.

Collaboration platforms associated to BPM (also called Social-BPM) meet relevant research aspects. The 'PROWIT' research project [45] financed by the German 'Bundesministerium für Bildung und Forschung' focuses on three important use-cases in context of business process management: collaboration in context of existing defined business processes, ad-hoc collaboration in context of undefined business processes and collaboration in context of process improvement projects. Philipp Walter [46] explains the different and innovative aspect of the PROWIT project:

We want to include every employee directly in the whole process life cycle instead of limiting him to process execution – even if he doesn't know how to read or modify business process models.

3.2 Business Process Modeling Formalisms and Standards

3.2.1 The Modeling Levels

Analyzing and modeling enterprise business processes helps identifying bottlenecks and taking strategic management decisions. From a semantic point of view, modeling means a transfer of knowledge into business objects. Business objects can be defined in different ways:

- Definition from the glossary of the University Information Service (UIS) of the Georgetown university:

  A physical or logical object of significance to a business; for example, a sales order, department, assembly, item, balance, or invoice. A business object is analogous to a class in object-oriented terminology.

- Definition from the glossary of the University of California Santa Cruz (UCSC):

  A concept from the everyday business terminology and vocabulary of the end-users. Each Object corresponds to a selection of data in the relational database, or a calculation or function using this data.

- Definition of Joseph Reddy, Software Engineer at iGATE global solutions:

  Business objects are programming structures that abstract the entities in the business that a program is written to represent. These entities should be recognizable to a non-technical person familiar with the business.

The three definitions denote a similar base comprehension of business objects, but on different levels:

- Conceptual level
- Logical level
- Physical level

These three levels can be compared to an MDA. This concept breaks the modeling activity down into three distinct model-driven steps. The figure 3.1 shows the different modeling levels and some of corresponding model types (for levels 2, 1 and 0).

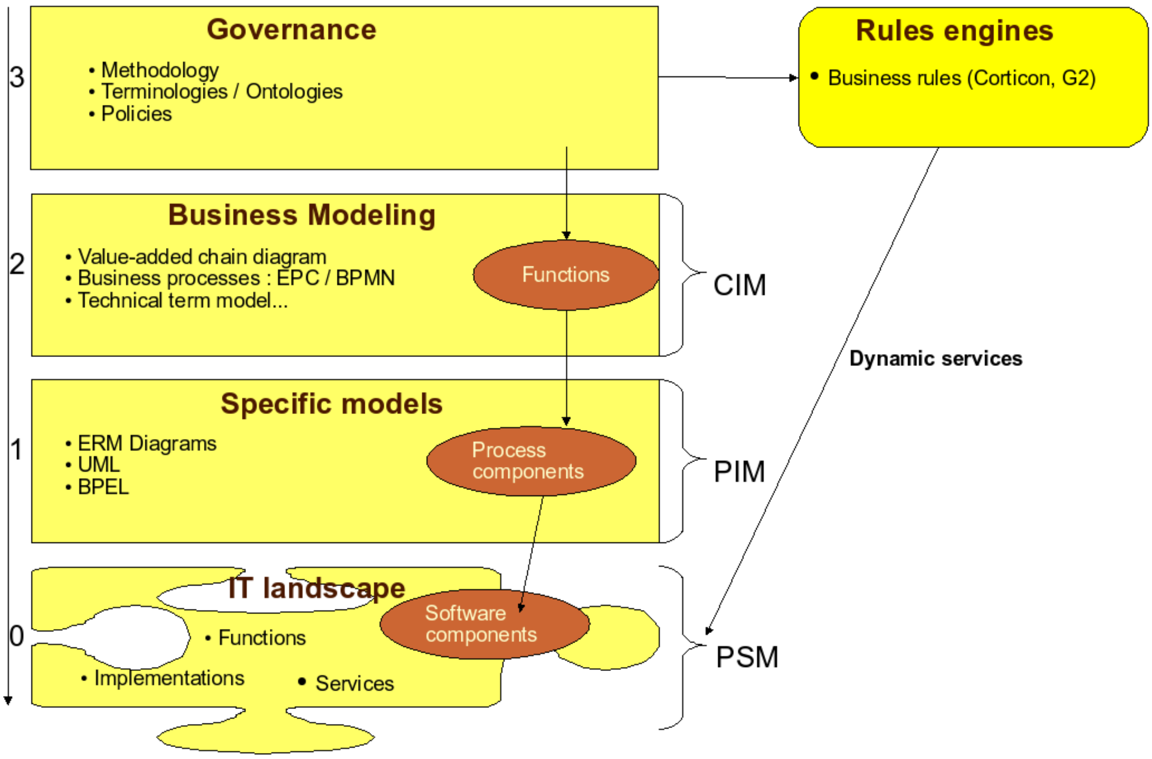


Figure 3.1: Modeling levels

- The first level (2), the CIM, represents the application's pure functional business logic without considering any computer-related aspects. From a business analyst's point of view, it consists of a set of models used for representing the company structure and process workflows without any implementation aspect. In the process life cycle, this level mainly corresponds to the design and modeling time.

- The second level (1), the PIM, provides a generic computing architecture, but not platform specific. A system architect or process engineer understands the business modeling and is skilled in information technology. Analog to a MDA approach, he is able, partly supported by automated mechanisms, to transform a CIM into a PIM.
- The third and lowest level (0), the PSM, represents the IT landscape of a company. It includes all the information needed to describe a deployable software system. Code and other development artifacts can then be derived directly from the PSM. A system analyst or software engineer, responsible for developing services or software components, is able to transform the PIM into a PSM, which includes the exporting of models which are near to execution like BPEL or BPMN models.

The levels 1 and 0 correspond to the process implementation phase of the process life cycle. In a real top-down MDA approach, the IT landscape should fit to the models established in higher levels. Practically, IT systems are mostly developed or introduced according to a spontaneous need without a real design from the business side. Sometimes business processes are even adapted to the IT processes. This approach is called top-down/bottom-up. In addition, legacy and redundant code are available and often deployed. This explains the puzzle form of the lowest level in the figure.

The highest level (3), called the governance level, defines all the corporate data, rules, policies and roles which must be considered, used and respected during the modeling activities.

To design, model, analyze and exchange a process, a common understandable language has to be employed. An appropriate formalism which acts as a standard for process modeling has to respond to following criteria:

- *Intuitive notation*: business process documentation is easier to understand and to analyze when it is represented in a graphical notation (like the common saying 'A graphic is better than thousand words').
- *Metamodel and vocabulary*: Defined objects and the relations between them. Appropriate metamodel instances for different modeling levels: value-chains, organigrams, IT levels and the possibility to navigate between the different levels.
- *Exchange format* for the processes and their diagrams.

Several standards or de-facto standards for business process modeling notations are developed by task forces or software providers. These notations coexist because they respond to different needs. None of them has been established as the only or perfect solution covering all the needs.

As a business process consists of one or several steps performed in a specific order, the notations must be able to represent workflows. Workflows representations are based on the mathematical graph theory, which use structures to model pairwise relations between objects from a certain collection. In most cases, the used graphs are directed graphs, allowing a hierarchical representation in organigrams or the time flow representation in business processes.

Håvard Jørgensen points out in his article '12 Different Ways to Model Business Processes' [47] that processes can be modeled in several manners:

Several languages have been proposed for business process modeling. Though most of them follow the conventional representation of processes as a series of steps, they emphasize different aspects of processes and related structures, such as organizations, products, and data. Consequently, they are suited for different kinds of processes.

The next section gives a deeper insight into the mostly used notations.

3.2.2 Business Process Modeling Notation

The first version of the Business Process Modeling Notation (BPMN) was created in May 2004 by the Business Process Management Initiative (BPMI). In June 2005, the BPMI has merged with the Object Management Group (OMG) to form the BMI DTF (Business Modeling & Integration Domain Task Force), or simply BMI. The OMG now maintains BPMN. The current version of BPMN adopted in January 2009 is 1.2 and the proposed one is 2.0, currently available as a second beta version.

The BPMN 1.2 specification [48] defines the goal of BPMN as follows:

The primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes.

The document 'Introduction to BPMN' [49] gives a high-level overview of BPMN. BPMN provides a set of graphical elements, divided in four basic categories:

- *Flow objects*: consist of the three core elements events, activities and gateways used to describe the main business process facts.
- *Connecting objects*: used to connect flow objects. The connecting objects consist of three types, sequence, messages and associations.
- *Swimlanes*: visual mechanism used to organize and categorize activities into pools and their containing lanes.
- *Artifacts*: allow to model additional information in a diagram to make it more readable.

The major drawback of BPMN is that it remains a very technical diagram type, which is not always mastered by business analysts.

3.2.3 Event-Driven Process Chain

The event-driven process chain (EPC) diagram was created in 1992 during a research project at the university of Saarland in Saarbrücken by a working group directed by the Professor August-Wilhelm Scheer in cooperation with the company SAP AG. The German books 'ARIS vom Geschäftsprozess zum Anwendungssystem' [50] and 'ARIS - Modellierungsmethoden, Metamodelle, Anwendungen' [51] describe the goals of EPC and their interactions with other diagram types.

The EPC diagram represents graphically a sequence of process steps in a directed graph. The triggers and the results of a business function are events. Functions and events are related with connections (arrows) and logical connectors (AND / OR / XOR). Additional information objects like 'document' or 'role' complete the process description.

The major drawback of EPC is that no released metamodel is available. The coverage of EPC elements provided in a modeling environment is let to the appreciation of modeling tools providers, who can extend the EPC syntax according to their needs or marketing strategies. However, the basic elements are always available. Due to the lack of normalized syntax and a poor support for technical aspects, the EPC cannot be automated as is. Some vendors are implementing transformations from the EPC to formalisms which can be executed.

While the OMG has created the BPMN standard, the EPC notation is often preferred by business analysts if they lack on technical background. EPC provides a higher level for modeling business operations, supporting technical terms in natural language. BPMN is oriented to execution and data elements are directly representing XSD elements. Moreover, BPMN models must respect a precise syntax while EPC can be human-readable and understandable even with incorrect structures. The article 'EPC vs. BPMN - the perfect flamewar' posted in the ARIS community [52] compares the BPMN and EPC formalisms in terms of control flow expressiveness and linking to other dimensions, with following conclusion:

There is no clear winner, because it always depends what you are looking for.

3.2.4 XML Process Definition Language

The XML Process Definition Language (XPDL) is an XML format standardized by the Workflow Management Coalition (WfMC) used to exchange business processes in a uniform way from a content point of view as well as from the point of view of the graphical elements. The WfMC addresses in particular the BPMN format [53]:

XPDL is the Serialization Format for BPMN.

The XPDL precursor called Workflow Process Definition Language (WPDL) was published in 1998. XPDL 1.0, based on WPDL, was approved by the WfMC in 2002. The current version XPDL 2.1, approved in April 2008, supports the BPMN 1.1. XPDL comes with an extension mechanism, which allows the definition of additional elements.

XPDL version 2.2 is currently in progress. Robert M. Shapiro [54] defines the following goals and features:

XPDL2.2 is intended as a preliminary release which supports the graphical extensions to process modeling contained in BPMN2.0. ... XPDL 2.2 provides a standard graphical approach to Business Process Definition based on BPMN graphics. XPDL 2.2 provides a standard file format for persisting BPMN diagrams and interchanging Process definitions.

3.2.5 Business Process Execution Language

The Web Services Business Process Execution Language (WS-BPEL), abbreviated Business Process Execution Language (BPEL), is an OASIS standard used to represent business processes in an XML formalism that is oriented to execution by using Web Services. BPEL does not have a generic visual notation, but some tool vendors are providing proprietary notation mechanisms. The first version called BPEL for Web Services (BPEL4WS) was submitted for standardization to the OASIS in April 2003. The current version of BPEL is 2.0 [55] and was released in April 2007.

BPEL is an orchestration language. Web Services associated to BPEL activities are orchestrated to create an executable process. The Web Services are exchanging WSDL messages. Those flows are defined by the BPEL process and controlled by the BPEL workflow engine. By definition, BPEL execution is limited to the Web Services techniques and does not consider human tasks, which are necessary in many business processes. The BPEL4People [56] is a specification which describes a BPEL extension which allows the execution of human tasks within the BPEL process. The extension based on the WS-HumanTask specification [57] introduces the definition of human tasks, including their properties, behavior and a set of operations used to manipulate human tasks.

BPEL, BPMN and XPDL are very complementary, with advantages and drawbacks. BPEL is an execution language that calls and synchronizes Web Services with no native support for diagrams. It requires technical details inherent to the Web Services technology, which makes it hard to deal with for business analysts. XPDL is a process design format storing even the coordinates of the objects placed in diagrams and is particularly adapted to the BPMN features. BPMN diagrams can be modeled visually without technical details at the beginning by business analysts, but must be provided later with technical details if it must be executed. Pierre Vignéras [58] makes a very critical analysis about the BPEL complexity and benefits, but this opinion is not shared by many tool vendors who have implemented or adopted BPEL workflow engines. In fact, BPMN, BPEL, XPDL and even EPC are coexisting and cooperating in many tools. The OMG provides an example-based recommendation set [59] to transform a BPMN process into a BPEL process.

3.2.6 UML Activity Diagram

The UML activity diagram is a workflow diagram which allows to represent a sequence of activities and flow control elements. Activity diagrams are generally used in software development to describe the underlying algorithm of a specific program part. Some business analysts or modeling environments are favoring activity diagrams for representing business processes. The advantage is that such diagrams are comprehensible by software engineers responsible for the implementation of business supporting components. In contrary, the closeness to computing languages makes it difficult to use by business analysts who do not have experience in software design. Other limitations are the missing resource-related or organizational aspects of business resources, like described in the evaluation article 'On the Suitability of UML 2.0 Activity Diagrams for Business Process Modelling' [60].

3.2.7 Static Model Types

Besides workflow diagrams and languages, business analysts are provided with a set of models which help them to describe the enterprise artifacts, organization and infrastructure. These models amend the previously described workflow models. This section describes the most common ones of them.

3.2.7.1 Added-Value Process Chain Diagram

The added-value process chain diagram is a high-level model which represents an enterprise as a chain of value-creating activities. Michael E. Porter [61] identified a set of generic activities common to many companies whose goals are to create a value which exceeds the production costs or the costs of providing the service and thus generate a profit. He distinguishes the primary activities from the support activities. The primary activities encompass inbound logistics, operations, outbound logistics, marketing, sales and service. The support activities encompass procurement, technology development, human resources management and firm infrastructure.

The top-level classification, which is usual today is originated from the EN ISO-9000 and following series of standards. The processes are divided into three categories: management processes, support processes and core processes. Each top-level category, in turn, is divided in subprocess categories. For example, a common classification for core processes is Customer Relationship Management (CRM) , Product Life cycle Management (PLM), Supply-Chain Management (SCM).

In the business modeling context, the added-value process chain diagram does not only describe the value chain, but serves as a high-level classification of business processes or parts of business processes. Elements of the diagram (modeled as functions) can contain links which lead to underlying processes. This thematically classification is also called 'Process map'.

3.2.7.2 Data Models

Data models allow a formal description of real-world objects which are relevant in a given context. In the business context, high-level data models, also called conceptual schema, semantic data model, information structure or logical data model, permit the establishment of glossaries of business terms which are involved in business processes like input or output data of given functions or role types of business process participants. The data model can define relationships between the business terms, but IT relevant aspects like access functions and operations are not addressed there. The so-defined objects are reused in business process models and as a blueprint for process implementation.

An Entity-Relationship Model (ERM) [62] is a data model specialization which is closer to software engineering. This model type includes entities which represent the object, relationships between objects with their cardinality and attributes associated to entities. The ERM model is particularly adapted for designing relational database schemas. The UML class diagram is widely used as an alternative to ERM-models, but is more adapted for the software class design implemented in object-oriented programming languages.

3.2.7.3 Organizational Charts

Organizational charts, also called organigrams, are diagram types which allow to design the structure of an organization on a geographical and human resources level as well as the relationships between the modeled entities. The geographical level encompasses the definition of organizational units like enterprise offices to distinguish the headquarters from the subsidiaries. Offices can be divided into departments and so on. The human resources level gives information about the management structure and the underlying employees with their responsibilities and relationships to the defined locations and departments.

Organizational charts are typically published within the company for communicating enterprise information. The main benefits are an effective collaboration between employees and the dissemination of information about organizational changes and management strategies. The modeled objects can be reused in business process models to link the organizational units, roles and people to given process steps and specifying their responsibilities or their involvement level. At process implementation level, the modeled people can be defined as executors or validators for given manual tasks, in particular when business processes are executed in a workflow engine.

3.2.7.4 IT Architecture Models

Business processes are supported by IT hardware and software or require the implementation/procurement of supporting components. IT infrastructure diagrams are used by business analysts to assess or design the infrastructure of IT hardware and system applications, the distribution of applications in a multi-tier architecture and the underlying protocols. System landscape diagrams are used to model the software components and sub-components of a given system and the interactions between the components. UML component diagrams are widely used as an alternative to these diagram types, but they are more intended to be modeled by software engineers or system architects.

3.2.8 Business Process Modeling and Quality Management Norms

Quality Management (QM) encompasses all management activities involved in providing products and services to customers which give them satisfaction. Quality management is divided in four stages: quality planning, quality control, quality assurance and quality improvement. Quality planning is a process which defines the quality goals and metrics as well as the methods and policies that will be used to achieve them. The quality control is a process of verifying the compliance of the produced products and services with the defined quality goals and avoiding as much as possible production defects or unsatisfying projects. Quality Assurance (QA) consists of procedures, processes and systems used to guarantee and improve the quality goals. Quality improvement comprises methods used to evaluate the quality management processes in order to improve them. These concepts and other quality management related aspects are explained in more detail in the book ”Project Quality Management - Why, What and How’ [63].

Since the 1980s, the trend moves more and more to Total Quality Management (TQM). TQM is a management philosophy which aims to continuously quality improvement to meet and exceed the customer expectations. Three steps are necessary to move from an acceptable level of quality (a calculated or historical level of quality which indicates the acceptance limits for products or services) towards a TQM:

- The processes are well-defined, audited and conform to the reference systems. The quality metrics of the products and services are defined by the company.
- The processes are evaluated according to the improvement objectives.
- The quality objectives are now measured from the customer and other involved parties point of view.

W. Edwards Deming [64] has following opinion about quality improvement:

Improvement of quality transfers waste of man-hours and of machine-time into the manufacture of good product and better service. The result is a chain reaction – lower *costs, better competitive position, happier people on the job, jobs, and more jobs.*

The link between QM and processes makes BPM a first choice management strategy to continuously improve the QM processes and increase performance. BPM helps to meet the requirements of International Organization for Standards (ISO) norms more quickly and thus successfully go through certification processes. Modeled business processes and the associated business repository can serve as a Quality Management System (QMS). The following sections give an insight of the most relevant norms in the QM context.

3.2.8.1 The ISO 9000 Family of Standards

The ISO 9000 family is a set of norms relative to QM [65]. The ISO 9000 official web page [66] describes the norm as follows:

The ISO 9000 family of standards represents an international consensus on good quality management practices. It consists of standards and guidelines relating to quality management systems and related supporting standards.

The ISO 9000 defines the fundamentals and terminology concerning QMSs.

The ISO 9001:2008 is the current standard which provides a set of standardized requirements for a quality management system. This is the only norm of the family which can be used as a base for certification. The Wikipedia page about ISO 9000 [67] gives an overview of the norm content. The main addressed topics are the requirements set to the management system including process documentation and quality manuals, the responsibilities of the management, the involvement of human resources, the procurement of non-human resources necessary for granting the product quality, the customer-related processes and the setting-up of measurement, analysis and improvement procedures to perform internal audits.

The ISO 9001:2008 is an abstract norm, which is written independently of the enterprise branch and domain of competence. It is why these norms are specified in more detail by domain-specific standards like:

- AS9000 (currently AS9100, revision C) [68]: Aerospace Basic Quality System Standard. An American norm containing 27 additional requirements which are specific to the aerospace industry.
- TL 9000 [69]: Telecom Quality Management and Measurement System Standard. An international norm specific to the telecommunication domain.
- ISO/TS 16949 [70]: defines the quality management system requirements for the design and development, production and, if relevant, installation and service of automotive-related products.

The ISO 9004 provides guidelines to move towards and implement a TQM.

The ISO 19011 provides guidelines for auditing QMSs and environment management systems.

3.2.8.2 The ISO 14000 Family of Standards

The ISO 14000 family concerns the environmental management. The norm addresses organizations which want to produce products and services with quality norms which respect the environment. The norm gives guidelines for setting up an Environment Management System (EMS). The official ISO 14000 web page [71] defines the norm's objectives as follows:

An EMS meeting the requirements of ISO 14001:2004 is a management tool enabling an organization of any size or type to:

- *identify and control the environmental impact of its activities, products or services, and to*
- *improve its environmental performance continually, and to*
- *implement a systematic approach to setting environmental objectives and targets, to achieving them and to demonstrating that they have been achieved.*

Even if enterprises apply this norm mainly on a voluntary basis, a high environmental quality production implies the continuous adaptation of the business processes.

3.2.8.3 Six Sigma

Six Sigma is a global management method which aims to increase the production quality by decreasing the amount of defects and the variability of the processes. Analytics and statistical methods are used to analyze the causes of the defects in order to improve the processes to avoid further analog errors. Performance controlling is a key point to stay at a high-quality level.

Thomas Pyzdek and Paul A. Keller [72] emphasize the link between Six Sigma and business processes:

SixSigma aims for virtually error-free business performance. Sigma, σ, is a letter in the Greek alphabet used by statisticians to measure the variability in any process. A company's performance is measured by the sigma level of their business processes.

The expected benefits of Six Sigma are the diminution of defects, trash, rectifications of rejects and amount of unsatisfied customers. It should lead to an optimal resources usage and as a consequence a reduction of the production costs to keep a competitive advantage.

3.2.9 Standards for Business Modeling Content

Predefined business modeling content consists of a set of standard reference models or recommendations created for a sector of activities. The reference models based on domain-specific best practices are describing business operations and relationships between business processes. As these processes are considered to be optimized for the addressed domain, the reference model providers recommend to the companies aligning their processes to the reference ones. The described processes can be transferred in business models (sometimes already part of the reference model) with respect of the defined terminology. Some modeling environment vendors are selling their own models, which were developed according to the specifications. This section gives an overview of the most common reference models.

3.2.9.1 Information Technology Infrastructure Library

The Information Technology Infrastructure Library (ITIL) is a set of concepts relative to IT Service Management. It provides good practices, which can be adapted by IT service providers or by the internal IT departments of all company types.

The [73] gives an executive summary about ITIL, including the expected benefits: a better alignment of IT processes to business needs, an increase of the effectiveness of the company and, in turn, a reduction of IT management costs. ITIL adopts a life cycle approach to IT services. The current version, V3, comprises five books, each of them dealing with one of the life cycle stages:

- Service strategy [74]: the IT service strategy must consider the requirements issued from the business strategy level in order to create IT services which support the business.

- Service design [75]: the 'fit for purpose' and 'fit for use' approaches ensure that the IT architecture and IT services are designed to meet the business objective of the customer or the company.

- Service transition [76]: managing and controlling changes into the live IT operational environment, including the development and transition of new or changed IT services.

- Service operation [77]: delivering and supporting operational IT services in such a way that they meet business needs and expectations and deliver forecasted business benefits.

- Continual service improvement [78]: learning from experience and adopting an approach which ensures continual improvement of IT services to better align to the business needs.

The ITIL glossary [79] provides the definition of the most relevant terms in the Information Technology domain.

3.2.9.2 Enhanced Telecom Operations Map

The enhanced Telecom Operations Map (eTOM) [80], also called Business Process Framework, is the most widely used and accepted set of standards in the telecommunication domain.

Kundan Misra [81] defines eTOM as:

a business process framework which places the service provider's enterprise within its overall business context. In addition to the operating processes involved in delivering telecom services, the wider enterprise of the service provider and its business interactions along with its relationships to other organizations are represented. Moreover, the eTOM describes the processes and interfaces that are required to insert the service provider's business into an e-business or e-commerce setting.

At the conceptual level, the framework is divided in three major process areas:

- Strategy, Infrastructure, and Product
- Operations
- Enterprise Management

The eTOM model is represented graphically as a set of interrelated rows and columns. Their intersections are the entry points to specific processes, which are organized in a hierarchical way. The hierarchy consists of 4 process levels (Level-0 to Level-3) with the Level-3 being the most detailed one.

3.2.9.3 Supply-Chain Operations Reference

The Supply-Chain Operations Reference (SCOR) is a process reference model endorsed by the Supply-Chain Council (SCC) addressing the supply-chain management processes. Process modeling, performance measurements with more than 150 defined KPIs and best practices are the core components of SCOR.

The overview paper of the SCC [82] defines following scope for SCOR:

- *All customer interactions, from order entry to paid invoice*
- *All product (physical material and service) transactions, from your supplier's supplier to your customer's customer, including equipment, supplies, spare parts, bulk product, software, etc.*
- *All market interactions, from the understanding of aggregate demand to the fulfillment of each order*

SCOR is based on five distinct management processes: plan, source, make, deliver and return. The efficient management of supply including inventory and shipping is necessary in order to increase overall product time to market.

3.3 Semantic Web Technologies

The World Wide Web (WWW) contains a huge amount of information, which is mainly only available in a layouted form. Raw data is rarely published by enterprises. The Semantic Web, which comprises a set of technologies which simplifies the data and metadata access to software agents, aims to solve this problem in a number of ways with the Linked Data method. The Linked Data design issue [83] is setting following requirements:

1. *Use URIs as names for things.*
2. *Use HTTP URIs so that people can look up those names.*
3. *When someone looks up a URI, provide useful information using the standards (RDF*, SPARQL).*
4. *Include links to other URIs so that they can discover more things.*

This concept encourages organizations to publish freely their data in an open and standardized way – using Semantic Web technologies – and to reuse data already available on the web. Tim Berners-Lee, the director of the World Wide Web Consortium (W3C), even introduced the term Global Giant Graph (GGG) in his blog [84] by referring to social networks. His vision of the 'Semantic Web' is a future where:

- Web information has exact meaning
- Web information can be understood and processed by computers
- Computers can integrate information from the web

Not only web applications benefit from Semantic Web techniques, but all applications which rely on underlying knowledge bases. In a key note session in 2002, he outlined the architecture of the Semantic Web [85] (see related figure 3.2). This architecture can be reused as a model for many application domains.
The next subsections describe the most relevant Semantic Web techniques.

3.3.1 Resource Description Framework

The Resource Description Framework (RDF) is a framework for describing resources in the web. RDF is a W3C recommendation issued from the W3C's Semantic Web Activity. The W3C [86] describes the features of RDF as:

RDF has an abstract syntax that reflects a simple graph-based data model, and formal semantics with a rigorously defined notion of entailment providing a basis for well founded deductions in RDF data.

RDF provides a simple data model and a syntax written in XML (called RDF/XML) that enable independent applications, parties and software agents to exchange and use

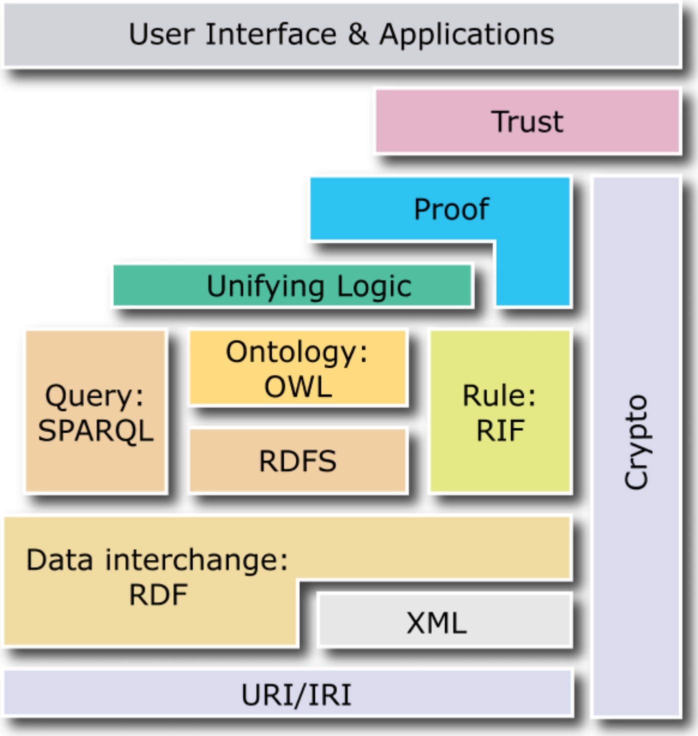


Figure 3.2: Architecture of a semantic web application

their RDF data. Some common applications using RDF are e-business applications, for example to query price and availability of shopping items or content description for search engines. The most famous RDF example is the Simple Dublin Core [87], which provides a standard set of fifteen properties used for describing web resources (e.g. Title, Creator, Subject, Description...) and intellectual property rights.

The underlying structure of any expression in RDF is a collection of triples, each consisting of a subject, a predicate and an object. Each triple represents a statement concerning the subject. The subject, also called a resource, is anything that can have a URI, such as a web page. The predicate, also called property, denotes a relationship concerning the subject. The object is the property value of the predicate.

Example: The document 'http://www.w3.org/TR/rdf-concepts/' is a recommendation of the year 2004. The following RDF document could describe this resource (namespaces are omitted):

```
<?xml version="1.0"?>
<RDF>
   <Description about="http://www.w3.org/TR/rdf-concepts" />
      <type>Recommendation</type>
      <year>2004</year>
   </Description>
</RDF>
```

The subject of the two statements above is: 'http://www.w3.org/TR/rdf-concepts'.

First statement:

The predicate is: 'type'.
The object is: 'Recommendation'.
The statement is: 'The type of document of http://www.w3.org/TR/rdf-concepts is a recommendation'.

Second statement:

The predicate is: 'year'.
The object is: '2004'.
The statement is: 'The resource http://www.w3.org/TR/rdf-concepts was published in 2004'.

RDF resources can be queried with the SPARQL Protocol and RDF Query Language SPARQL [88], an official W3C recommendation. SPARQL allows the querying of an RDF graph with an SQL-like syntax, including conjunctive patterns, value filters, optional patterns, and pattern disjunction. The SPARQL protocol supports the remote invocation of SPARQL queries via HTTP or SOAP.

The RDF Schema (RDFS) version 1.0 specification [89] describes how to use RDF to describe RDF vocabularies and defines other built-in RDF vocabulary initially specified in the RDF Model and Syntax Specification.

The RDF's XML format is considered to be cumbersome when expressing and reading manually RDF statements. Three RDF serialization alternatives having human-readability as a key requirement are established: Notation3 (N3), Terse RDF Triple Language (TURTLE) and N-TRIPLES.

- N3 is a non-XML serialization format for RDF models which is more compact and readable than RDF/XML, but more expressive. The N3 specification [90] defines the N3 language as follows:

  This is a language which is a compact and readable alternative to RDF's XML syntax, but also is extended to allow greater expressiveness. It has subsets, one of which is RDF 1.0 equivalent, and one of which is RDF plus a form of RDF rules.

- TURTLE [91] is a subset of the N3 language, but a superset of the N-TRIPLES format. TURTLE limits its syntax to the RDF graph model. TURTLE has no official standard or recommendation status, but its simplicity and human-readability makes it a first choice alternative for Semantic Web developers.

- N-Triples [92] is a line-based and plain-text serialization format for RDF and a subset of TURTLE. It was designed to be the simplest of the three alternative formats, but missing features make it loose on expressiveness, which, in turn, increases its complexity when managing manually large amounts of data.

3.3.2 Web Ontology Language

The Web Ontology Language (OWL) is a formal description for creating, publishing and distributing ontologies. Ontologies are a collection of domain-specific terms that are classified according to a taxonomy. A taxonomy is a classification model or schema used to categorize objects with specific criteria in classes.

OWL is a specification of the W3C , which historically based on the no longer developed Darpa Agent Markup Language + Ontology Inference Layer (DAML+OIL) language. The current version of OWL called 'OWL2' refers to the specification of 2009 [93] while the name 'OWL' refers to the specification of 2004 [94].

OWL is built on top of RDF, but its expressiveness is greater than RDF-S. It provides constructs which allow predicate logic. OWL is conceived with three language levels:

- OWL Lite: lowest level with the highest restrictions, used to describe simple taxonomies.
- OWL DL, where DL stands for 'Description Logic': OWL DL is more expressive than OWL Lite, but still has restrictions to ensure decidability. Decidability is ensured when an algorithm can define for each element of the ontology whether it has a given attribute or not. A typical restriction is that a class is not allowed to be an instance of another class.
- OWL Full: contains no restrictions and allows high predicate logic description.

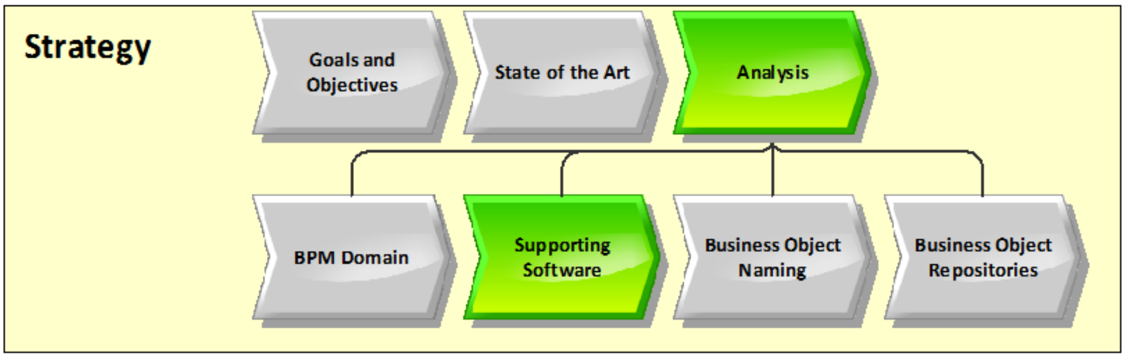
Many tools are streamlined for OWL DL and Lite due to the simpler languages constructs which allow complete inference.

The language syntax provides a set of predefined classes adapted to ontologies like the class 'owl:Class' used for describing taxonomies or the root class 'owl:Thing'. Instances, called individuals, are objects belonging to classes. Each object or class can be specified in more detail by properties. Datatypes properties allow to associate simple types like XML schema datatypes to classes or individuals while object properties describe relations between classes. Operators like unions or intersections allow the description of enumerations or cardinality.

To favor the interpretation of OWL ontologies, the Semantic Web proposes two additional formalisms. The Semantic Web Rule Language (SWRL) is a W3C proposal [95] which defines a syntax for rules used to formalize knowledge. The rules describe the implications between antecedents and consequences. Antecedents are described in the rule body (condition part) and the consequences in the rule head. SWRL is not the only ontology rules description language. The Rule Interchange Format (RIF) is a W3C recommendation [96] defining a format used to transform one rules description language into another one to favor interoperability.

Chapter 4

Employed Software and Methods



This chapter gives an overview of tools and methods used to write this dissertation, model the example processes and develop the prototype. For each product, the official website is referenced. All provided links were successfully accessed on December 15, 2010.

4.1 Text and Images Processing

4.1.1 TeXShop

The dissertation is written on a Mac OS X version 10.4.11 (Tiger) operating system with the latex editor 'TeXShop' version 2.14. TeXShop is a free TeX editor and previewer for Mac OS X developed by an American mathematician, Professor Richard Koch. TeXShop is licensed under the GNU GPL. The official website is 'http://pages.uoregon.edu/koch/texshop/'.

4.1.2 Pencil

The graphical user interfaces which are part of the user concept (see chapter 6) are developed with the open source prototyping tool 'Pencil 1.2' for Windows 7. The Pencil Project comes with built-in stencils for diagramming and prototyping. It is licensed under the GNU GPL. The official website is 'http://pencil.evolus.vn/'.

4.1.3 Paint.net

The images were processed with the free image and photo editing software 'paint.net' version 3.5 for Windows 7. The interface is quite simple, but the features are more powerful than the Microsoft Paint software, which comes with Windows. The official website is 'http://www.getpaint.net/'

4.1.4 NeoOffice

NeoOffice is a full-featured set of office applications, including word processing, spreadsheet, and presentation programs for Mac OS X. The office suite was created by taking the features of Oracle's OpenOffice.org office suite and adapting them to needs of Mac users. It results in a very stable code with increased performance. The official website is 'http://www.neooffice.org/'.

The spreadsheet application of NeoOffice was used to create the graphics related to the statistical results.

4.2 BPM Software and Methods

The dissertation is illustrated with business process model examples. Additional UML diagrams are supporting the development concept of the prototype. The modeling activities require a business process modeling environment. The main diagram types required are:

- Organigram diagram
- IT infrastructure diagram
- Data model diagram
- Value-added process chain diagram
- Event-driven process chain diagram
- BPEL diagram
- UML class diagram
- UML component diagram

The modeling environment must also provide features which allow modeling a SOA. The last requirement is of technical order. To support the statistical analysis activities through the dissertation, a business repository serving as a corpus is available in form of ARIS databases. Tools able to read this format and export appropriate data are necessary. All the requirements are met by two tools provided by the Software AG company (previously by the IDS Scheer AG for many years), which is in leading position in the BPM market.

4.2.1 ARIS Express

ARIS Express is a free modeling tool providing nine basic model types, including the major part of the required model types with the exception of BPEL and UML models. The business objects available for each model type are the most common and most frequently used object types, but represent only a subset of those available in the professional modeling environments of the vendor. The drawback of the free version is that the models are stored in independent files. Business objects cannot be reused like in a business repository and models cannot be linked together. The target user groups are BPM beginners who want to familiarize with modeling concepts and occasional users. The official website is 'http://www.ariscommunity.com/aris-express'.

Whenever possible, the free ARIS express version 2.2 was preferred for modeling the diagrams included in the dissertation. ARIS express models can be distinguished from the other ones from a look and feel point of view: the colors are more intensive. In the user concept chapter, ARIS Express is displayed in the background of some of the screen mock-up to represent an arbitrary modeling environment.

4.2.2 ARIS SOA Architect

The UML models and the models necessary for exposing the SOA use case are not available in the ARIS express version. The professional version of ARIS, called ARIS Platform, is a set of products covering the IDS Scheer's approach to BPM projects: Strategy, Design, Implementation and Controlling platforms. All ARIS products are conceived as a three tier (three layers) distributed client/server architecture. The three layers are the presentation layer containing the graphical user interface (GUI), the business layer containing the business logic and the data layer containing the ARIS database, called the business repository. The figure 4.1 gives an architecture overview of the products of the ARIS platform.

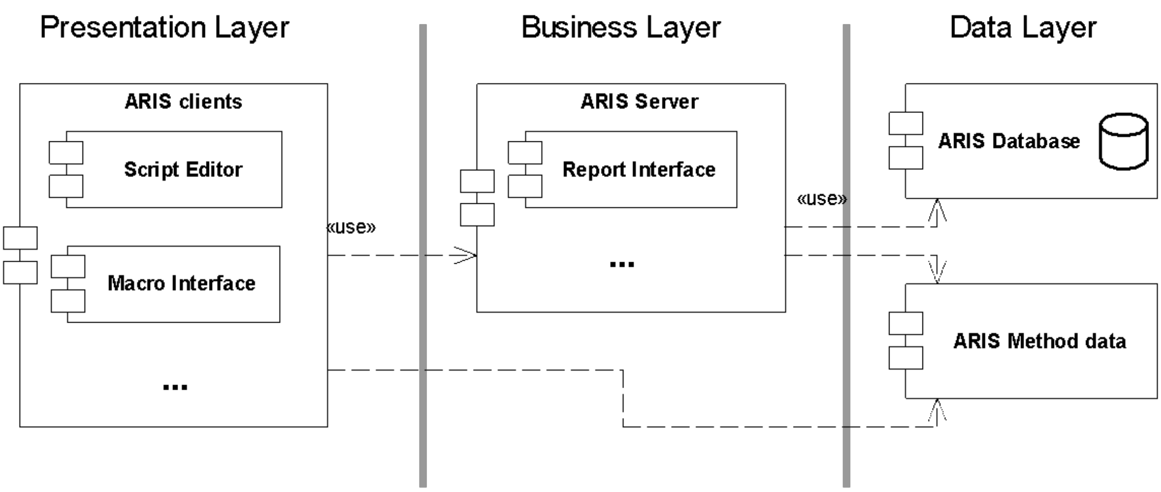


Figure 4.1: ARIS software architecture

ARIS comes with script interfaces, allowing the users to access the business repository via scripts, written in JavaScript or in visual basic. The macro interface acts on objects which are available on the client side, while the report interface only deals with objects which are stored in the business repository. A script wizard enables users to create their own report scripts allowing the customized automation of many functions previously performed manually. The output can be generated in several different formats (XLS, DOC, RTF, TXT, HTML, PDF). In this dissertation, the script interface facilitates the export of selective content of the corpus business repositories in order to perform the study.

The product 'ARIS SOA Architect 7.1' anchored in the implementation platform covers all of the current modeling needs. It contains the methods for process design which are also available in ARIS Express (EPC, Value-added Chain diagrams...), but amended among other object types with SOA specific objects. Additionally, this product offers methods for process implementation, including the BPEL, BPMN and UML method. From a functionality point of view, it provides checks for validating the structure of process models and an automatic transformation from EPC to the BPEL formalisms. The official website of the ARIS platform is 'http://www.softwareag.com/fr/products/aris_platform/'.

4.3 Software Development Tools

Each feature described in this dissertation is implemented and tested as a prototype on a Mac OS X platform. This section describes all the software engineering tools involved into the development work.

4.3.1 The Eclipse Integrated Development Environment

'Eclipse' is a free Integrated Development Environment (IDE) mainly written in the JAVA programming language that allows the software development in each programming language. The particularity of this environment is its flexible architecture, which allows the integration of plug-ins to extend the functionalities. The official website of the eclipse community is 'http://www.eclipse.org/'. The prototype was developed with the eclipse version 3.3.2.

4.3.2 The Eclipse Modeling Framework

The Eclipse Modeling Framework (EMF) is a modeling framework and code generation facility for building tools and other applications based on a structured data model. The model specification is described in XMI. From the model, EMF provides tools and runtime support to produce a set of Java classes for the model, a set of adapter classes which enable viewing and command-based editing of the model, and a basic editor. EMF provides the foundation for interoperability with other EMF-based tools and applications. The official website is 'www.eclipse.org/emf/'.

4.3.3 openArchitectureWare

The openArchitectureWare (oAW) is a modular MDA/MDD generator framework implemented in JAVA. It supports parsing of arbitrary models, and a language family to check and transform models as well as generating code based on them. Supporting editors are based on the Eclipse platform. The oAW has moved to Eclipse as a part of the modeling project. The version (4.3.1) used in the dissertation is still available at the website 'http://www.openArchitectureWare.org/'.

4.3.4 The JAVA Programming Language

JAVA is an object-oriented and interpreted programming language released in 1995 by Sun Microsystems, today Oracle Corporation. The JAVA platform allows the development and the execution of programs written in JAVA on many operating systems and processors. The main component of the platform, the JAVA virtual machine is implemented for each target system and should guarantee the same behavior of a JAVA program on each system. The portability of JAVA code is one of the main advantages of this language. Numerous frameworks and APIs are available so that JAVA can be used for many different purposes. The official website of the language is 'http://www.java.com/'.

The version used for developing the main components of the prototype is the JAVA standard edition 5 (J2SE 5) for the Mac OS X Tiger operating system. The current version JAVA 6 is available since 2006, but is not compatible with Tiger.

4.3.5 The PERL Scripting Language

Perl is an interpreted language created in 1987 by Larry Wall. Perl is a scripting language which is particularly adapted to the treatment of text files because powerful regular expression features are fully integrated into the language syntax. The official website of the Perl language is 'http://www.perl.org/'.

Through the dissertation, the Perl language supports many statistical tasks, especially when information had to be extracted from text files. The used version is Perl 5.8.6, delivered with the Tiger operating system.

4.3.6 MySQL Database Management System

MySQL is a relational database management system licensed under the GNU General Public License as well as, depending on the company needs, under proprietary agreements. MySQL was owned by the Swedish company MySQL AB and bought by Sun Microsystems in 2008, now owned by Oracle Corporation. MySQL runs under many operating systems and the databases can be accessed via several programming languages, including JAVA. The official website is 'http://www.mysql.com/'.

In this study, the free MySQL version 5.1 is used, mainly to store the content of the corpus business repository after the natural language processing in a structured way (see section 4.5) in order to facilitate statistical evaluations.

4.4 Knowledge Representation Software

4.4.1 FreeMind

FreeMind is a free mind-mapping tool written in JAVA. The version 0.8.1 of FreeMind is used in the dissertation in order to insert mind maps figures which summarize or highlight some facts. The official website of FreeMind is 'http://freemind.sourceforge.net/'.

4.4.2 The Protégé OWL Editor

The Protégé OWL editor is a free, open source ontology editor and knowledge-base framework. It allows users to create and manage their ontologies in a visual way, particularly in the OWL format. The official website of Protégé is 'http://protege.stanford.edu/'.

In this study, the Protégé OWL editor version 3.3.1 was used because it was a stable release meeting the requirements during the development stage. For the automation of OWL data management, the JAVA API coming with this version is employed.

4.5 Natural Language Processing Software

In this study, the tasks related to the controlling of business object labels or, more generally, to natural language analysis is based on the idea that the use of linguistic intelligence as a pre-process simplifies and powers subsequent processing steps. With linguistic intelligence, natural language related tasks are realized, such as morphological, syntactic and semantic analyses, as well as translation related tasks. An off-the-shelf Linguistic Engine (LE) applying to the German language and meeting following requirements is necessary:

- Delimitation of sentence boundaries and word boundaries.
- Morphological analysis which reduces the words to their base form.
- Delivery of a set of linguistical, grammatical and semantic information about words.
- Delivery of a citation form.
- Recognition of different country variants.
- Tagging of unknown words.
- An acceptable state of runtime performance and robustness which allows its actual deployment in industrial scale settings.

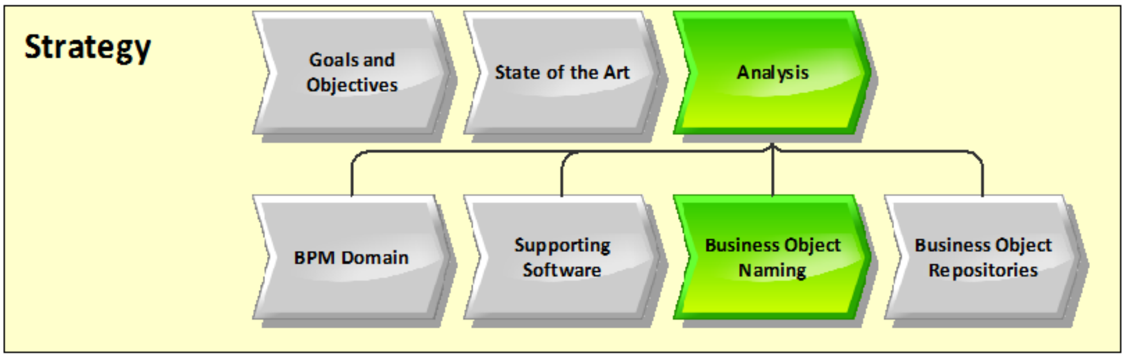
The LE provided by the IAI e.V. (Institut für Angewandte Informationsforschung) Saarbrücken fulfills the requirements. The IAI has recently concluded a joint venture with the company Across to create a common company, the 'Congree Language Technologies GmbH'. The official website is 'http://www.iai-sb.de/iai/'.

The LE is implemented as an unification-based approach, which favors runtime performance. This approach allows the broadest experimentation width and the maximum validation of different deep linguistic processing steps in terms of information richness on the levels morphology, syntax and semantics, which are accomplished by the single operation unification. The LE tokenizes words and assigns generic linguistic information to them. A chunking pre-process serializes the input into meta-information about the processed resource and LE-relevant information. The employed linguistic resources of the LE, mainly dictionaries, are additionally customized for specific categories of entities and subject fields. Thus, the originally assigned attribute-value information of the LE can be adapted qualitatively by using internal and external contextual information (refinement through performance control), e.g. user dictionaries and revised annotations based on statistical models such as terms in context, which have been identified in separate learning phases.

The LE produces output in form of feature bundles, with each bundle containing the generic linguistic information for a token. The linguistic information is coded as attribute/value(s) pairs. The most common attributes are listed in the appendix B.

Chapter 5

Life Cycles of Enterprise Processes



The combination of Business Process Modeling and Business Process Management is one of the major methods used to continuously adapt the given processes and workflows of an enterprise to react on new business needs, environments and requirements. A large set of the business objects that occur in process models are labeled with natural language expressions. These object labels are normally neither checked in terms of their language competence nor in terms of their business performance. The whole process management life cycle is affected by issues concerning natural language items. In the business process management domain, several life cycle models are commonly used like the one described in the section 2.2. A more fine-grained life cycle model is preferred in order to better assign and classify the identified problems. The figure 5.1 shows the adopted process life cycle stages:

The following sections describe the problems encountered during the different process life cycle stages excepting the process controlling. In this phase, the processes are monitored to find out bottlenecks during the process execution. The natural language issues have no direct impact on the controlling phase, but the monitoring metrics, in particular for the execution of manual processes, can prove the concept validity.

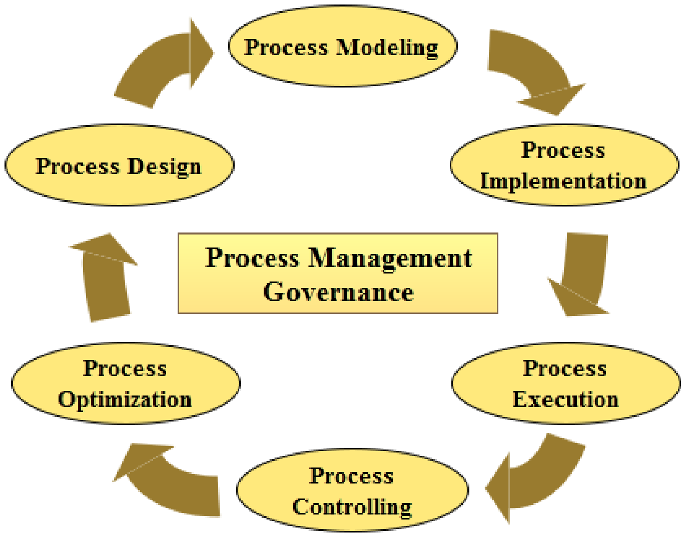


Figure 5.1: Adopted process life cycle stages

5.1 The Business Process Governance Level

Process governance means the process of processing process management. At this stage, several strategic decisions concerning the **formal methodologies**, the **workflow of process management** and **content guidelines** have to be taken.

The **formal methodologies and policies** define how the processes should be modeled like the following aspects:

- Used BPM software
- Model types used to describe the processes
- Modeling structure and checks to verify the structure at formalism level
- Rules for generating dynamic services

Formal methodologies are widely used and proofed. BPM software includes a set of tools that check the model structure and the formal structure compliance. It concerns rules like 'A process starts with events and ends with an event'. Formal methodologies will not be the point of interest.

The **process management workflow** is a process defining the different steps a process/business analyst must perform during the whole process management life cycle. This process can differ according to the process management objectives. This chapter describes the most common workflow steps including the impact of the business object denominations.

Content guidelines are setting up quality metrics for business object denominations contained in models in terms of correctness, appropriateness and readiness. Guidelines recommend naming conventions for business objects and the appropriate usage of corporate terminologies, ontologies and thesauri. Several situations can occur:

- No terminological resources are available at all
- Terminological resources are available, but they consist only in a collection of terms gathered from different sources like external domain terminologies, databases used for localization purposes or even a collection of terms which once had occurred in the company. This kind of resources is typically plagued with redundancies.
- In rare cases, clean and valuable terminologies are available

According to experiences in the domain of controlled authoring, the quality of a terminology checker used to control a document written in a given language correlates with the quality of the underlying terminological resources. However, organizations are rarely ready to invest in building or cleaning their terminologies.

5.2 The Process Design Phase

In the process design phase, the business analysts prepare the modeling tasks by analyzing the existing processes and designing the to-be processes according to the management strategy. Process modeling requires a deep knowledge of the company or department structure and workflows. Existing internal documentation provides the first source of knowledge for business analysts, like word files, text documents, semi-structured XML documents or even from graphical formats like Visio, which are often stored in a CMS or DMS.

If process documentation is inexistent, understanding and structuring the processes require mainly human interactions. As a first step, business experts are interviewing the people who are able to describe the existing processes. The interviews are performed by following specialized interview sheets. The main questions, which are expected to be answered from process documentation or interview sheets, are:

- *What*: What is the described task/activity?
- *Which*: Which capabilities (automated or manual) are required by the task?
- *Who*: Who is the person, the role, the department, the organization unit responsible for the task?
- *When*: Which event triggers the process step? Is the event a periodical event?
- *How*: Is the task a manual or an automated task? Which software or hardware system infrastructure is supporting the task? Is the process a candidate to be supported by some standard definitions or implementations?
- *Input or output*: Which are the objects processed by the process step? What is the expected result of the process step?

- *Decisions*: Is the result of the process step a yes/no decision? What are the following decision paths?

Current BPM tools do not support the modeler to bridge the gap between existing documentation/interview sheets and the process design. Furthermore, process models are reflecting the reality only for a limited time span because a company has to adapt its processes to stay reactive and competitive. Often changes are only registered in documents and not in process models or reciprocally. No tool is able to detect the conflicts between changes in the documentation and the modeled processes. In other cases, a process modeler may even fail to know where to find the appropriate documentation in the CMS or DMS.

Process design does not always start from scratch. Due to the life cycle structure of the process management stages, the design phase can also serve as a starting point of process optimization in the context of process change management. If process models are already available, the business analyst must be able to find the concerned processes in the business repository in order to improve them according to new requirements. Impact analysis is a specialized search procedure that finds out which processes can be impacted from a process step modification. Findability issues are explained in more detail in the next sections.

5.3 The Process Modeling Phase

Once the design questions are answered, the process can be formalized with BPM tools. Modeling the processes with a BPM tool can be considered as a knowledge transfer from unstructured information into a model-based structured business repository, where the business entities and enterprise workflows are stored in the form of business objects related with connections. The business modeling environment provides to the business analyst a set of models, objects, attributes and connection types which have specific semantics (see section 3.2). These objects are named with real world artifacts or with string literals (organization names, department names, software components...). This section gives an example of a standard modeling use case and outlines the problem types occurring in business object labels.

5.3.1 The Business Process Modeling Workflow

5.3.1.1 Description of the Enterprise Structure and Architecture

As a first step, a business modeler often starts by describing the enterprise organizational structure (see 3.2.7.3) because this gives him key information like people responsible for departments of interest or who is performing the processes he currently analyses. The figure 5.2 shows an example of an organigram with its main objects: 'organizational unit', 'location', 'role' and 'person'. The object names are a mix of natural language items and string literals like person names or loose department names.

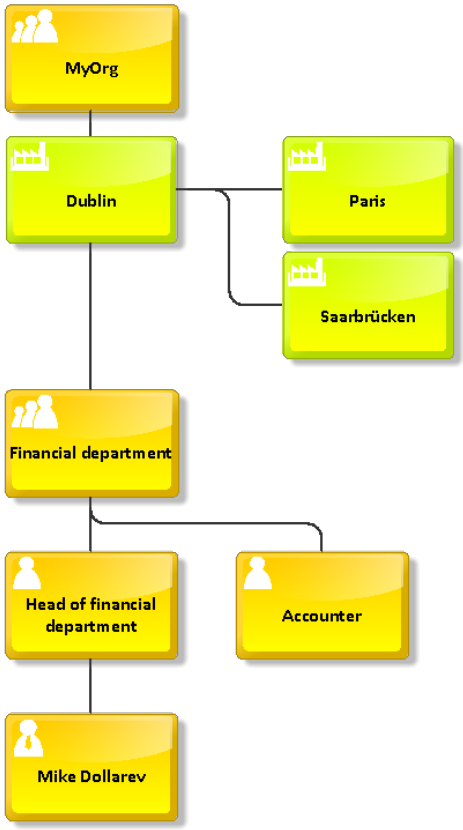


Figure 5.2: Organigram

The IT infrastructure including hardware and software is the backbone of business support. Business analysts parse and describe the IT infrastructure with the 'IT infrastructure' and 'system landscape' model types (see 3.2.7.4). The figure 5.3 gives an example of software systems supporting the Customer Relationship Management (CRM) modeled in a 'system landscape' diagram. The object names are string literals denoting software system names and are not part of natural language items.

Enterprise software systems and business processes are processing input and output data. The model type 'data model' (see 3.2.7.2) is created on the business side and gives additional information about relations between business objects. The data description may be used as a template for generating an item in a programming environment embedded in a MDA. The technical terms are natural language items, sometimes including abbreviations or string literals defined in other business object types. The figure 5.4 shows an example of a customer data object with its unique identifier 'customer identifier' and some personal information. This object can be used by components of the CRM system landscape and modeled as input or output data in a process model.

Depending on the diagram types and modeling methods available in the modeling environment, additional static diagrams can complete the enterprise description. For

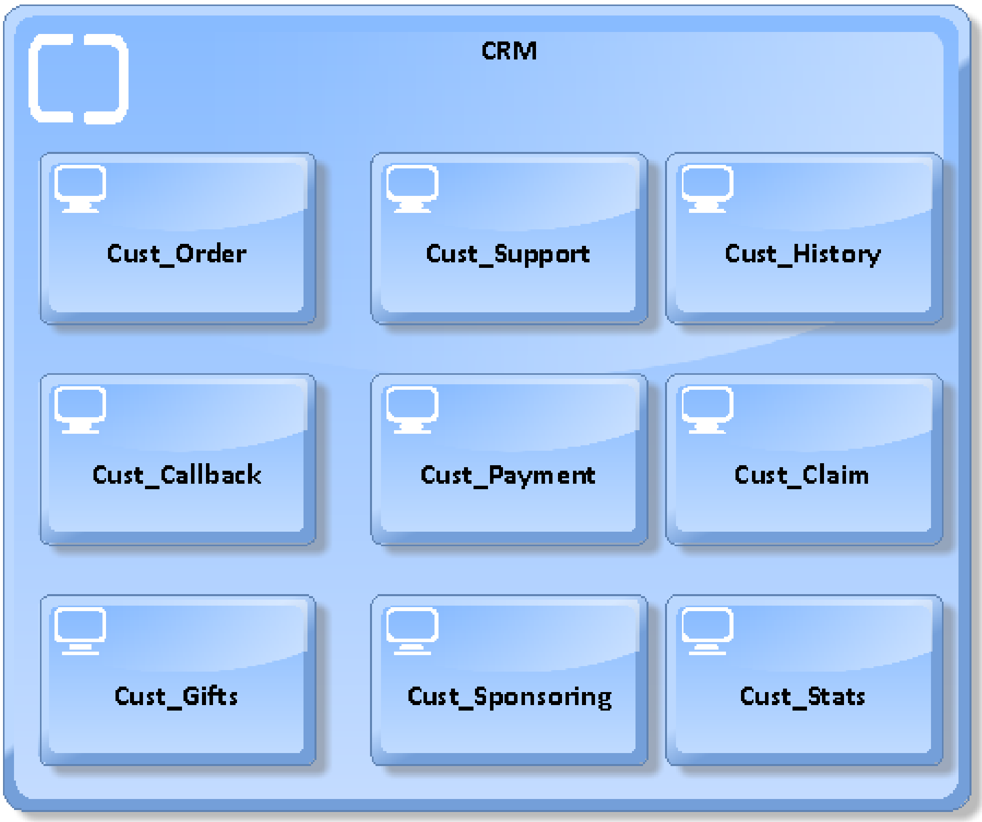


Figure 5.3: CRM system landscape

example, a product tree diagram can describe the products or services offered by the company, a market diagram can describe which markets are currently available or are part of future objectives...

5.3.1.2 Description of the Enterprise Processes

Once the static models are created, the business analysts start to model the business processes themselves. A process consists of one or several steps, which can be divided in sub-processes. Each sub-process can be modeled in a different model and connected to the according process step to avoid getting one huge and confusing process model. This modeling strategy results in a process hierarchy, where the highest model is a coarse-grained model mostly used for process navigation purposes and where the deepest models (often 'leaf' models) are fine-grained models that are candidates to automation.

The model type 'added-value process chain' acts as an entry point to the process models. It describes a process map of the organization (see the section 3.2.7.1). The figure 5.5 shows a process map subset including some supporting processes. While modelers mostly agree with the widely used entry points 'Management processes', 'Core processes' and 'Support processes', no standard process map is provided by business process modeling software, even for a given domain. Some of the best practices

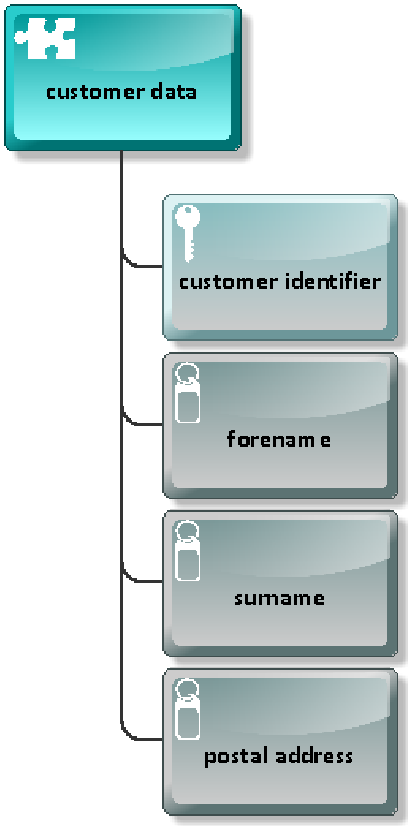


Figure 5.4: Data model

result in a process map which can be reimported in further projects, but not in an established map included in modeling environment. Process maps are prone to the same natural language issues as well as other model types even though a certain degree of domain-specific standardization may be here possible.

At this point, the business analyst models the process workflows and associates the highest level processes to a function modeled in the process map. The methods used at this step are mostly BPMN (see section the 3.2.2) or EPC (see the section 3.2.3) because they have become standards for business modeling due to their fit for business requirements, both technical and enterprise. In all subsequent examples, the EPC will be preferred since the formalism requires less technical modeling items and thus is more intuitive for business analysts. The quasi standard EPC comes with the following main object types, which respond to the questions gathered during the design time:

- Activity (also called process step or function): Responds to the question 'What is the task?'
- Requirement: Responds to the question 'What are the requirements of the task?'
- Organization units, people, roles: Respond to the question 'Who performs the task?'

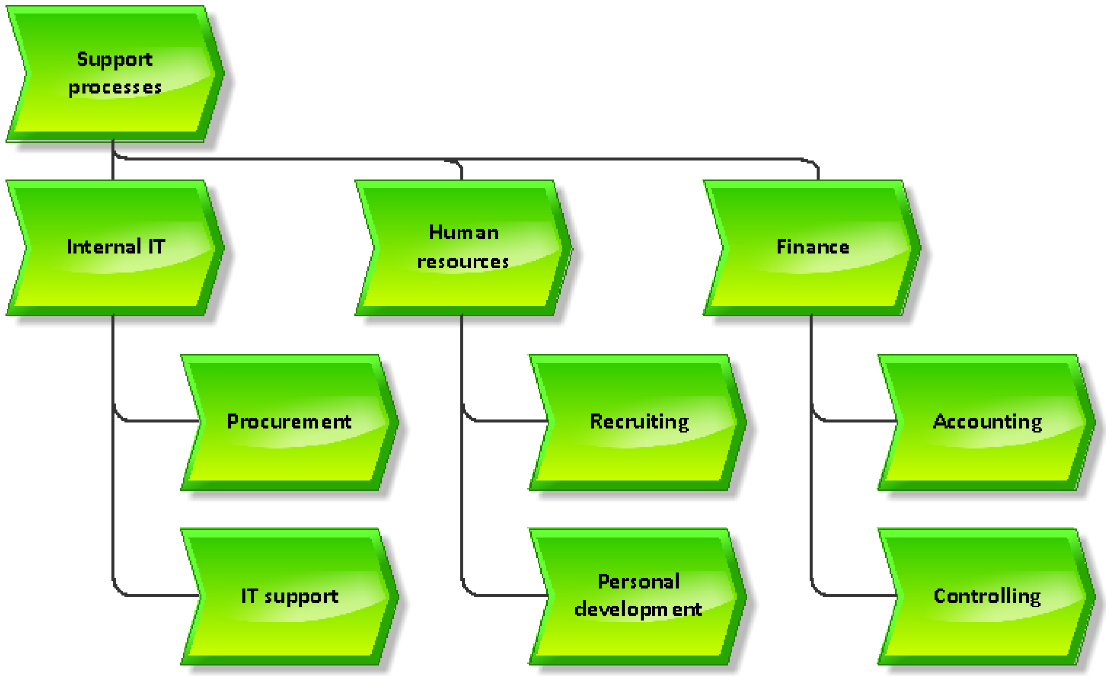


Figure 5.5: Process map subset

- Events: Respond to the question 'When is the task performed?'
- IT systems: Respond to the question 'Which system supports the task?'
- Technical terms / clusters, representing the input and output data: Respond to the question 'What input and output objects are concerned by the task?'
- Connectors (and / or / xor), to model the decision paths

The figure 5.6 shows an English EPC model concerning the order of an online product (electronic book, software...). The figure shows a sequence of events and functions related with connections and, when needed, with 'xor' connectors. The turquoise object ('customer data') is the data object previously modeled in the data model example. The light blue objects ('Cust_History', 'Cust_Stats', 'Cust_Support') are software components supporting the process steps. They were previously modeled in the CRM system landscape model. The yellow organization unit ('IT support') that is supporting the manual process step 'contact customer' was previously modeled in the organigram.

The figure 5.7 shows the same EPC model than the figure 5.6 but translated into German.

5.3.2 Object Naming Issues

A fair amount of business objects modeled in the example models of the previous section are natural language items. When the model is saved, all business objects,

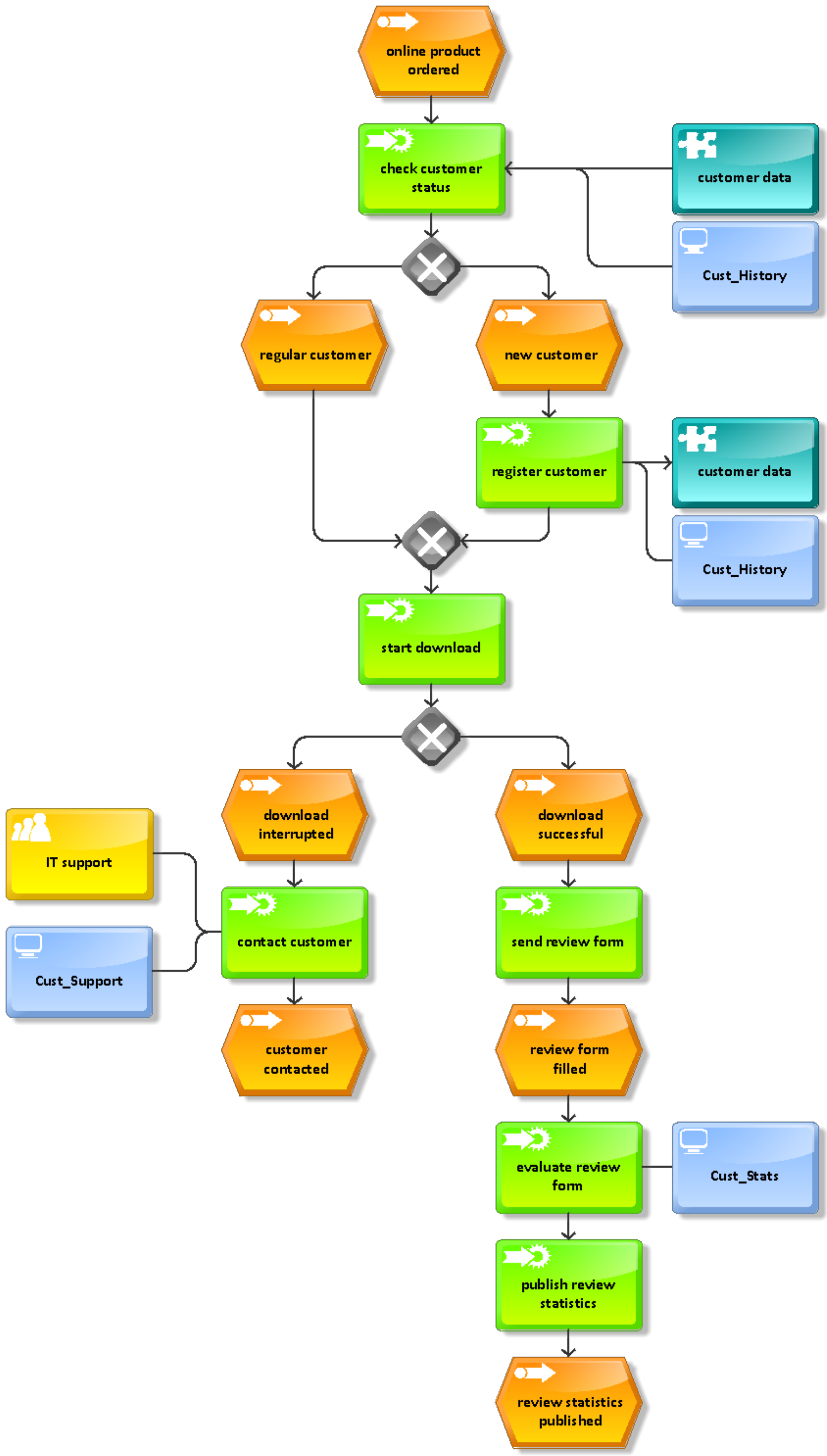


Figure 5.6: Online order EPC

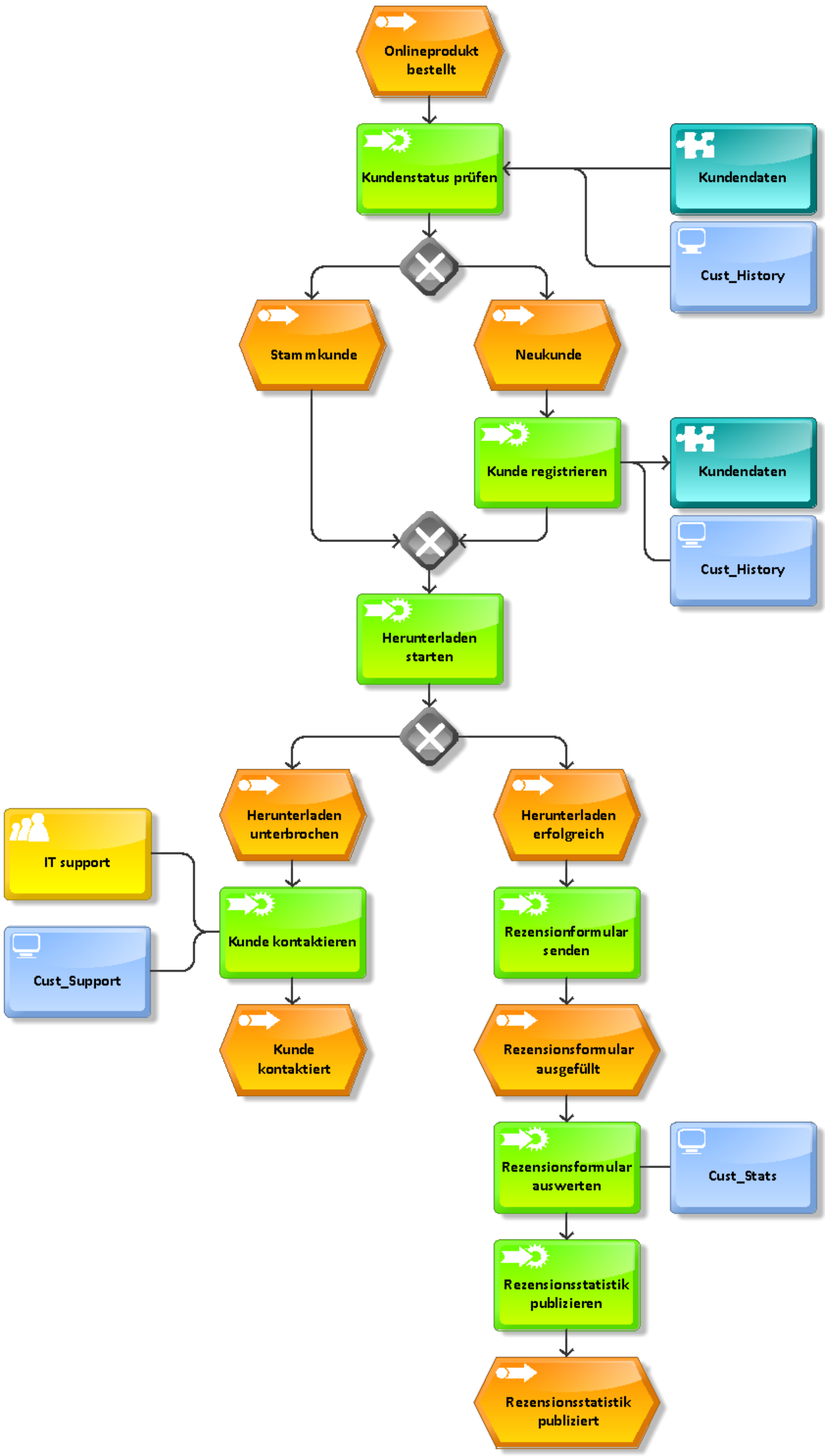


Figure 5.7: Online order EPC

connections and attributes are stored in a business repository. To ensure findability and interoperability of business processes (nearer described in following sections), each created business object should occur only once in the business repository. Object redundancies have to be avoided. Current modeling tools are already notifying if an object with the same name is existing, in order to reuse this one as reference. However, if two object names differ in only one character, the BPM modeling environments can't detect the variants. It leads to unwished inconsistencies and redundancies in the business repository. A major problem of modeling environments consists in the lack of support for the object naming in terms of correctness, appropriateness and readiness.

The inconsistencies are due to different causes:

1. Spelling and grammatical issues (correctness)

2. Choice of the terminology (appropriateness and coherence)

3. Non-respect of the naming conventions for the different object types (readiness)

Natural language issues occur in each chosen modeling language. In global companies business models may be translated in each subsidiary language, which duplicates the described problems. However, the next examples will mainly be in German: although multilingual issues are addressed, the scope of this concept concentrates on a monolingual solution in German language. The developed techniques can then be adapted for other languages.

5.3.2.1 Spelling and Grammar

Spelling and grammatical checking ensure the correctness of business object denominations. Common spelling errors are mostly caused by typing errors when using a computer keyboard. Typical error types are double letters, letter rotations and missing letters. A common spelling checker can help finding out these errors, but correction propositions will not be adapted to business modeling vocabulary.

Another type of errors are due to the regional languages variants like British vs. American English (for example 'Initialisation' and 'Initialization'), or for the German language, the variants from Germany, Austria or Switzerland. At the governance level, the company must define which kind of spelling variants must be used during modeling. If several countries with same language setups are involved in the process, regional variants must be detected to avoid redundancies in the business repository.

The recent spelling reforms, which aim to unify the spelling rules for all German speaking countries are an additional source of errors in the German language. The new rules concern amongst others hyphenation rules, whether words should be written separately or as a compound word or the spelling of foreign words. The DUDEN [97] gives a short tutorial concerning the new spelling rules. Fact is that different kind of spelling rules could coexist in business repositories and these inconsistencies have to be detected as well.

Concerning grammatical issues, standard grammar checkers are best working on sentences. Complete sentences are unwished in business object denominations. Hence, grammar rules have to consist only of a subset of standard grammar rules to avoid false warning alarms.

5.3.2.2 Terminology Usage

A correct terminology usage ensures the appropriateness of the business objects denomination to avoid misunderstandings between process participants and to facilitate the transformation in an IT solution (see the section 5.4.5). While common spelling errors could be detected by integrating spelling checkers in BPM tools, no automatic check is available for verifying the business terminology usage. To develop models which are compliant to the corporate terminology (if available!), the business analyst must have a deep knowledge of the terminologies and an accurate attention during modeling. This deep knowledge is still more difficult to be ensured when the modeling is performed on a consultancy basis from an external company. The resulting inconsistencies are amplified in a multi-user modeling environment. Data objects created in a data model like the figure 5.4 are reused in events and functions in business process models (BPMN, EPC). If a terminological resource is available, the data objects should reflect it. Otherwise, data models are the only terminological resource to consider when modeling business processes. No check ensures that data objects created in data models are reused in function and event objects occurring in BPMN or EPC models.

The data object names consist in one or more nouns mostly in singular, sometimes completed by an adjective.

Examples of German data objects:

- 'Beschlussvorlage'
- 'Bewilligungs-Richtlinie Großkredite'
- 'neue Konditionen'
- 'Kompetenz-Richtlinie'

The terms used as data objects are source of inconsistencies even if they are correctly spelled. The following examples outline the main causes of inconsistencies:

- *Hyphenation*: 'Kundenauftrag' / 'Kunden-Auftrag'. The general hyphenation rules that apply to business terms should be defined on governance level. Appropriate checks must be provided to ensure the rule compliance. The quality assurance model developed in this dissertation will recommend a set of default hyphenation rules. Authoritative terminology entries may stand in contradiction with the corporate hyphenation strategy.

- *Abbreviation usage*: 'Kd. Auftrag'. Business analysts might use non-authorized abbreviations instead of the full and precise term.
- *Synonym usage*: 'Kunde' versus 'Auftraggeber'. Parts of compound terms can be synonyms: 'Kundendaten', 'Auftraggeberdaten'
- *Genetive form usage*: 'Verwaltung der Daten' instead of 'Datenverwaltung',
- *Grammatical variant*: 'das Verwalten', 'die Verwaltung'

As the data object names are determining the information object names in functions or events, their names have to be precise. In the following example, inconsistencies are due to an imprecise term usage. Unlike the examples above, the imprecise term can not always be mapped unambiguously to the corresponding term.

'Kundenauftrag' or 'Produktionsauftrag' instead of 'Auftrag'

If no terminological resource is available at all, a set of standard and domain-specific business terms can be very useful to fulfill the naming requirements. The same counts for available terminologies if the term entries are not customized for business needs.

A correct term usage still does not ensure the coherence of the business object denomination. For instance, an order can be 'created', 'canceled' or 'processed' but can not be 'called' or 'phoned'. An order can have the status 'new' or 'available', but not 'yellow'. No checks are currently available to verify the correct association between business terms used in a business object name.

5.3.2.3 Naming Conventions Compliance

Readiness is ensured when naming conventions that care for simplicity and clearness are respected. The non-respect of naming conventions is an additional factor that makes the business repository grow inconsistently. Since no official standard for business objects naming is existing, BPM tool vendors recommend a set of object-specific conventions, which are widely accepted by the business modelers and taught in specialized modeling training sessions. Such conventions should be adopted at governance level. Since no automatic checks are provided by BPM tools, business analysts have to pay attention to the naming rules when modeling.

The next two paragraphs outline the basic naming conventions for activities and events that will be a starting point for developing a quality assurance model for business object denominations.

Basic guidelines for process steps: The process step denotes the action that has to be performed at a given time. Following basic rules must be observed:

1. The name of a process step must consist of an information object and an activity. The information object is a noun, mostly in nominative singular. The activity is a verb in the infinitive form.

- ’Kundenauftrag freigeben’

2. The information object must reuse the precise term modeled in the glossary models and defined in the terminology. This point corresponds to the appropriateness issues described in 5.3.2.2.

   - ’Kundenauftrag freigeben’ instead of ’Auftrag freigeben’

3. The order of noun and verb must be respected. Common source of inconsistencies are due to inverting the noun and the verb.

   - ’Freigeben Kundenauftrag’
   - ’Entscheide Kredit’

4. Substantivation has to be avoided.

   - ’Kundenauftrags-Freigabe’
   - ’Freigabe Kundenauftrag’

5. The verb ’durchführen’ has to be avoided.

   - ’Kundenauftrags-Freigabe durchführen’ instead of ’Kundenauftrag freigeben’

Basic guidelines for events: Events are describing the result of a process step. Following basic naming rules must be observed:

1. The event name consists of an information object and a status modification. The information object is a noun, mostly in nominative singular. It corresponds in the majority of cases to the information object occurring in the previous process step. The status modification is an adverb, and adjective or a verb conjugated in the perfect participle form, often corresponding to the infinitive verb occurring in the previous process step.

   - Kundenauftrag freigegeben

2. An event can be formulated as a negative event that consists of the positive formulation and the negative mark ’nicht’. A common source of inconsistencies consists in using alternatively a positive or a negative formulation:

   - ’Akzeptiert’: –> ’Nicht akzeptiert’ / ’Abgelehnt’

3. The previous rules do not always apply if the triggering process step delivers a decision result. A decision is composed of at least two events. Their names reflect the results of the process step.

   - Bonität OK
   - Bonität nicht OK
   - Kundenwunsch = Hotelreservierung

4. The unnecessary usage of an auxiliary verb is forbidden. This case is commonly found in customer business repositories.
   - 'Rechnung ist erstellt' instead of 'Rechnung erstellt'
5. The usage of the particle 'zu' must be avoided since it denotes a future event. Events are always results and do not introduce future actions.
   - 'Kundenauftrag ist freizugeben'

The table 5.1 shows examples of inconsistent event objects:

Correct event	**Variant 1**	**Variant 2**	**Variant 3**	**Variant 4**
Antrag angelegt	Antrag ist angelegt	Antrag wurde angelegt	Antrag wird angelegt	Antrag kreiert
Antrag nicht angelegt	Kein Antrag angelegt	Antrag ! angelegt	Antrag ist nicht angelegt	
Antrag i.O.	Antrag in Ordnung	Antrag OK	Antrag ist i.O.	
?	Datei soll gelöscht werden	Datei muss gelöscht werden	Datei ist zu löschen	Datei zu löschen

Table 5.1: Example of inconsistent events

5.4 The Process Implementation Phase

The presented process implementation scenario is based on the expert paper '10 steps to a business-driven SOA' [98], which is a typical use case implemented in the 'SOA Architect' tool (see 4.2.2) for bridging the gap between modeled business processes and the IT implementation in a SOA. The objective is to associate services to activities defined in a high-level process model and to transform the business process in an automated process that runs in a dedicated workflow engine.

5.4.1 Service Discovery Issues

In the first step, the business analyst creates the high-level process. The figure 5.8 shows a small EPC concerning stock exchange processes. This step is still done during the modeling phase.

The business analyst can model the input and output data on the function. In the figure 5.9 the technical term 'Aktienverkaufspreis' is the result of the function 'Aktiendaten abfragen'.

The objective in this SOA scenario is to find an appropriate service which fulfills the function requirements. For this purposes, the business analyst can annotate the process

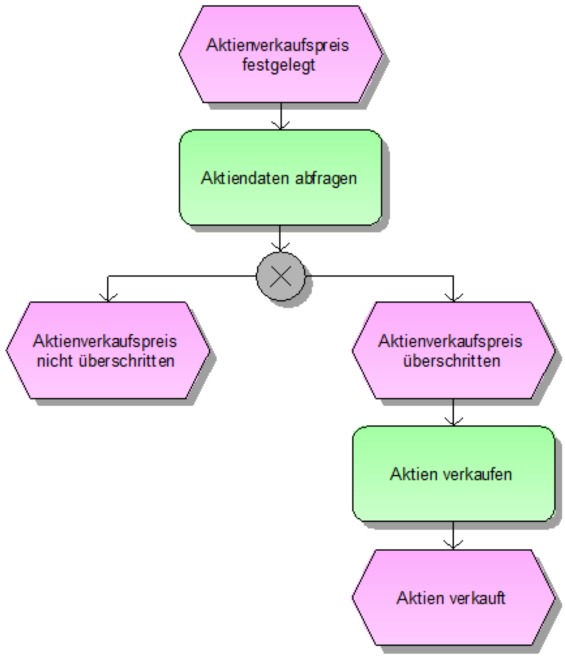


Figure 5.8: German EPC

step with a requirement object. Requirements can be of two types: functional and non-functional. Functional requirements describe the need on a service functionality level, e.g. what the service should do. Non-functional requirements are general conditions or policies that are independent of the service functionality like encryption, high-availability, performance. The figure 5.10 shows a functional requirement modeled on the 'Aktiendaten abfragen' process step.

In a SOA, the service requirement and data modeled on activities need a counterpart on service level. The service must be annotated in an analog way to find a corresponding service. The figure 5.11 shows the expected result: a software service (StockExchangeDAX) is connected to the activity 'Aktiendaten abfragen' to enable further automation steps.

To find a service, the service must be visible, either for a manual search through service registries or for discovery by software agents.

The OASIS reference model defines following prerequisites:

For a service consumer to discover a service, the service provider must be capable of making details of the service (notably service description and policies) available to potential consumers; and consumers must be capable of becoming aware of that information.

To associate a service to an activity, the business process modeler must dispose of a set

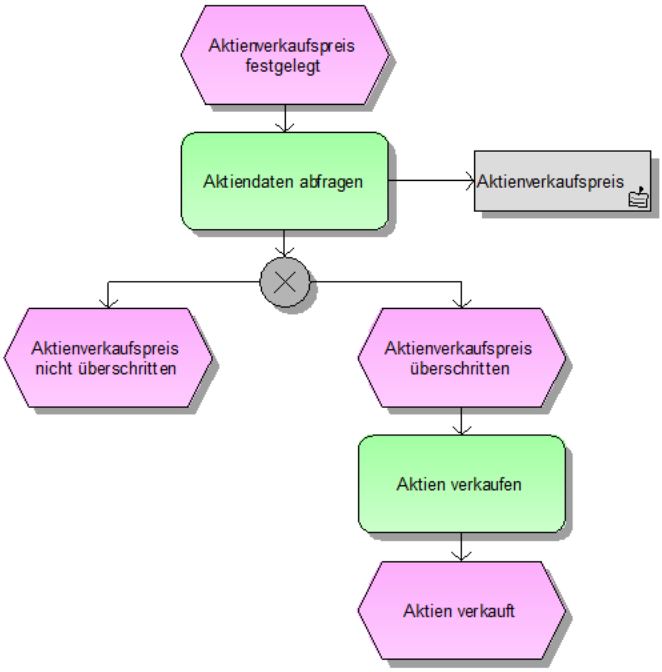


Figure 5.9: EPC with technical term

of business objects that represents the technical service in the modeling environment, here called 'software service'. BPM software comes with a business repository into which technical services are imported and mapped to 'software service' objects for an intuitive representation, which can be reused for modeling purposes. In the most SOA modeling environments, the imported services are WSDL-files because of their available technical interface using the SOAP protocol (see the section 3.1.2.2). The difficulty resides in the fact that the appropriate software service has to be found by process modelers who are not aware of the technical functionalities of the service. The service discovery solution adopted in this SOA scenario is to describe the software services in specialized models in two different ways: structurally and semantically.

5.4.2 Structural Description of Services

Structural description of services means that annotations are needed on a technical level. When importing the service, technical elements are also stored to the business repository as specific objects like function interfaces and function parameters. The data objects previously modeled as input or output data on the process step must be mapped to the parameters of the software service on the IT level.

The figure 5.12 shows the mapping between the data object 'Aktienverkaufspreis' and the IT input parameter of the software service represented by the 'class' object 'GetLastSaleSoapIn'. This class object is an input parameter for the Web Service 'StockExchangeDAX'.

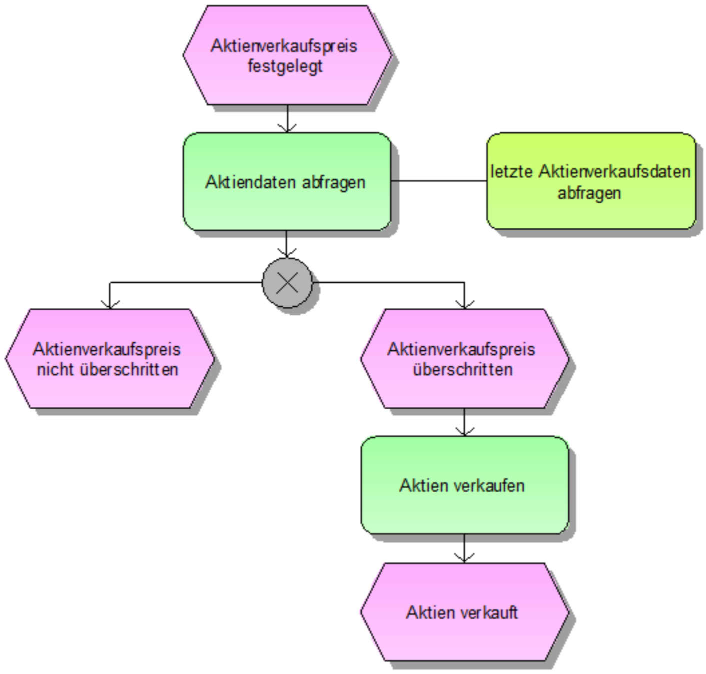


Figure 5.10: EPC with requirement object

The service connected to the activity in the figure 5.11 can be a result of a structural service search. This kind of description brings some inherent problems:

- The business analyst or process engineer needs IT skills to be able to map the data objects to the IT structures.
- The mapping is hardly coupled to specific protocols (here SOAP).
- The services must be imported before mapping, so that all the technical structures are available for the mapping procedures. As a consequence no automatic search over service registries is possible before examining the IT structure.
- The technical term glossary must be exhaustive enough to dispose of all needed objects.
- The naming issues described in the previous section (inconsistencies) also apply to the service description: if a synonym or a less specific term is used for describing the service, no match can be found between the data modeled on activities and data modeled on software services.

5.4.3 Semantic Description of Services

To avoid the problems caused by a structural service search, another strategy is to describe the service semantically. In the process model, requirement objects are connected to the activities. The requirement object expresses in natural language

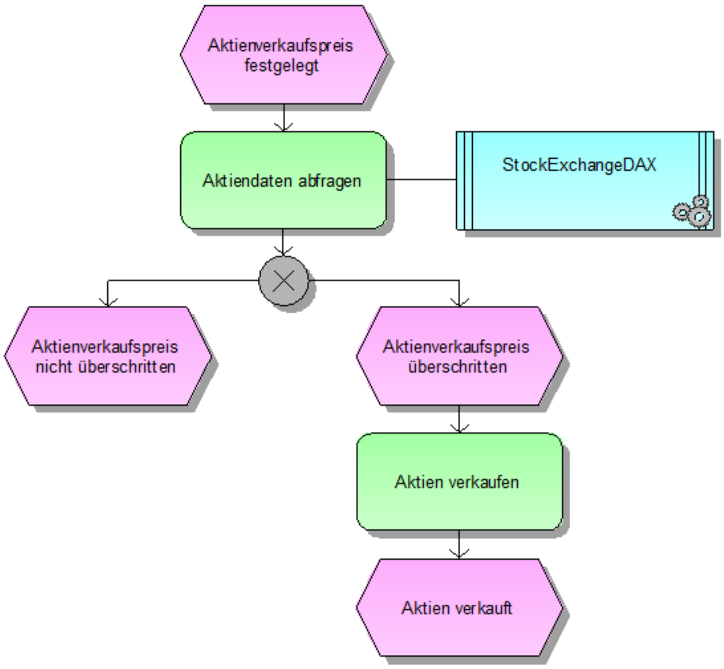


Figure 5.11: EPC with a software service

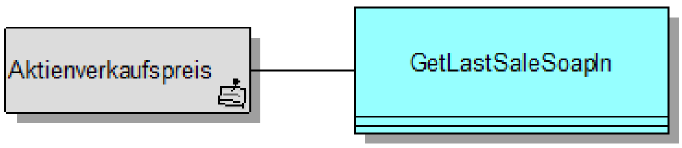


Figure 5.12: Output data mapped to the service parameter

the functionality of the needed service like previously seen in the figure 5.10.

To find an appropriate service based on the semantic description, the requirement objects must be connected to the software service. In this case, the requirement object is called capability, which means that the service is capable to fulfill the requirements. The figure 5.13 shows the requirement 'letzte Aktienverkaufsdaten abfragen' connected to the software service object 'stockExchangeDAX'. The UML component sharing the same name is a technical representation which allows the exportation of a service blue print for further implementation in developing environment. This object is not important here.

In addition, this approach is problematical in different ways. To allow the matching of requirements and capabilities, it leads to double modeling efforts: requirement objects must on the one hand be related to the activity on business side to describe the needs and on the other hand on the IT level as capability to describe the software service functionality. Both objects must be identical. Generally, different persons are involved in the modeling process. For instance, the business side may be modeled by a business analyst and the IT side by an IT architect. In this case, it is not ensured that the

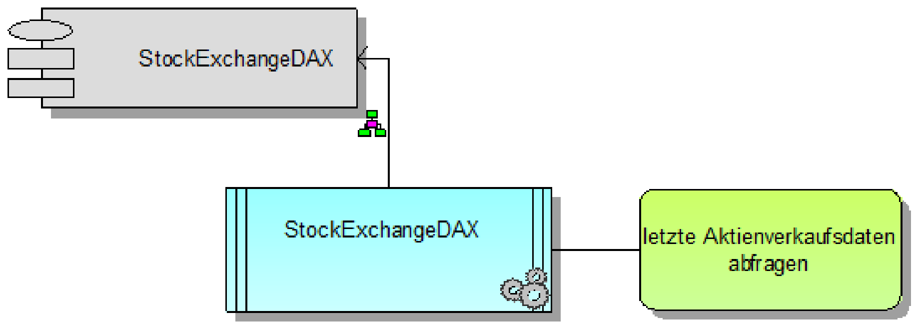


Figure 5.13: Software service annotated with capability

two or more persons are modeling the same appropriate capabilities or use the same granularity for the description. The naming issues like inconsistencies prevent from an effective service discovery. If only one person is responsible for modeling both parts, the modeler could directly connect software services to the business function and ignore the requirement/capability modeling. However, this approach would facilitate service discovery over service registries if the capabilities/requirements objects are correctly labeled.

Web Services often come with a description included in a 'description' tag in the WSDL formalism. This tag outlines the service capabilities in natural language. Modelers can be inspired from that description to create the capabilities objects, but the description is not automatically exploited.

To facilitate Web Service discovery, recent research efforts tend to Semantic Web services, that are enriching Web services with machine-processable semantics. The Web Service Modeling Ontology (WSMO) from the W3C provides a conceptual framework and a formal language for semantically describing all relevant aspects of Web services in order to facilitate the automation of discovering, combining and invoking electronic services over the Web.

The W3C describes the goal of the WSMO framework [26]:

WSMO provides ontological specifications for the core elements of Semantic Web services. In fact, Semantic Web services aim at an integrated technology for the next generation of the Web by combining Semantic Web technologies and Web services, thereby turning the Internet from information repository for human consumption into a world-wide system for distributed Web computing.

This approach is very similar to the capability description, but the idea is to supplement the service description with a set of ontological concepts that can be automatically interpreted by machines. Unfortunately, the concepts must be known by the service provider as well as by the service consumer side to assure an effective collaboration between the involved parties, so that the annotation problems remain the same.

5.4.4 Service Abstraction Level

The structural and semantic service description procedure is hardly coupled to technical services like Web Services. However, in a SOA different types of services coexist: manual tasks, hardware-supported services or software services implemented with different protocols. If a software service which supports an activity fails, another service or a manual task could temporarily replace the failing service. The same objects must be modeled on each service which fulfills the same requirements and are processing the same input and output data. This is why a degree of abstraction is introduced in the service description: an object representing an abstract service.

The model 5.14 shows an abstract service 'Börsenservice'. Capabilities and data objects are related to it.

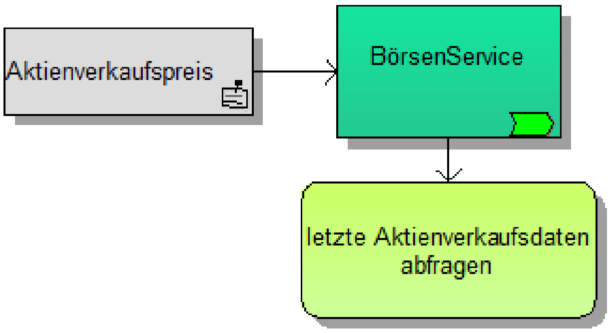


Figure 5.14: Abstract service annotated with capability and data object

The model 5.15 shows that the service 'StockExchangeDAX' fulfills the criteria of the abstract service 'BoersenService'. All services corresponding to the modeled data and capabilities can be connected to the same abstract service. This results in a collection of possible services which can support a given task.

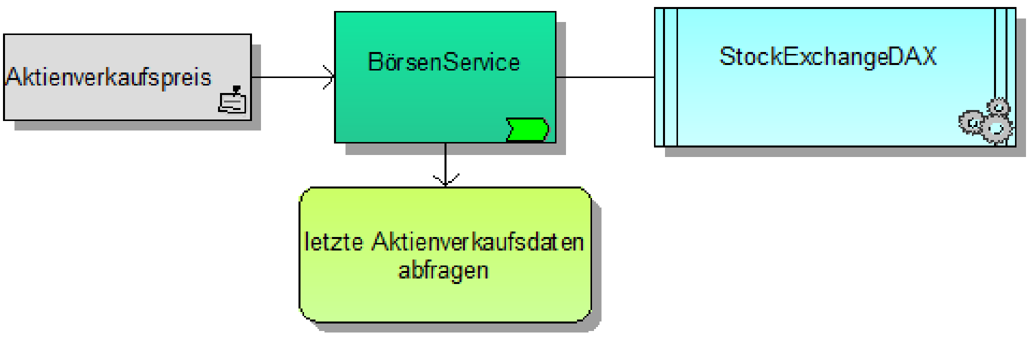


Figure 5.15: Abstract service associated to a software service

The EPC extract 5.16 shows an abstract service connected to the activity.

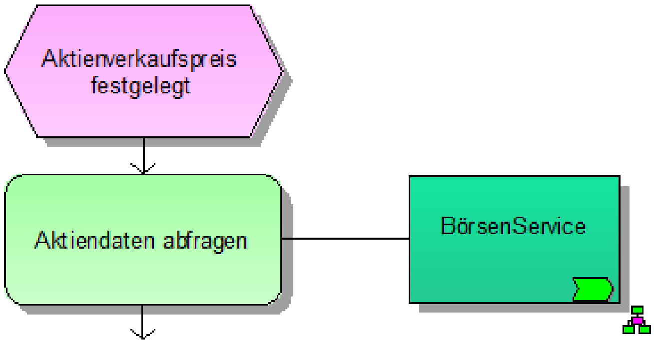


Figure 5.16: Abstract service associated to an activity

When searching a service, a list of services related to the abstract service is given and the appropriate service is connected to the activity. The EPC 5.17 shows that the software service 'StockExchangeDAX' was found as an adequate service.

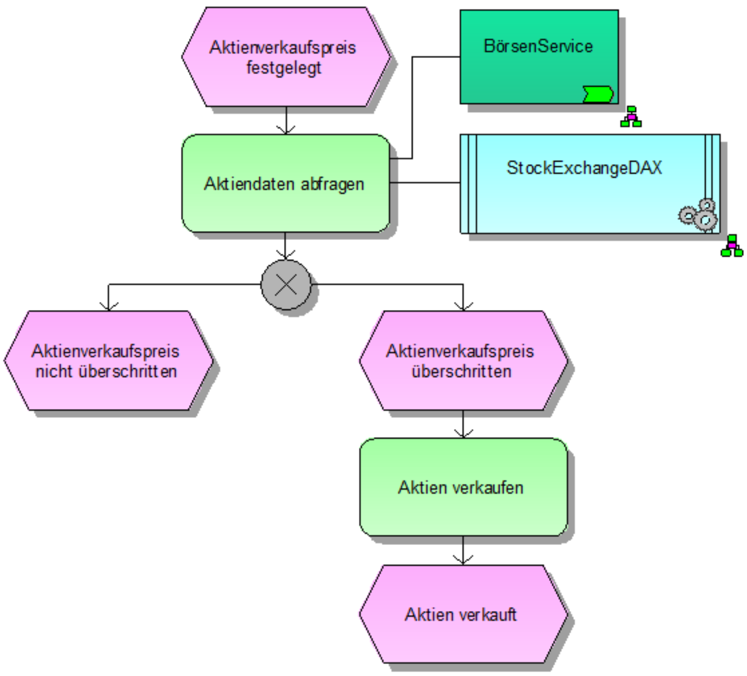


Figure 5.17: Abstract and software service associated to an activity

In theory, the service abstraction level can eliminate a certain redundancy when describing analog services. But in fact, the difficulties of the previous structural and semantic descriptions are only moved from the activity to the abstract service. Software (and other) services must still be annotated when importing the service interface in order to associate them to the correct abstract service, as well as activities must be annotated to find an abstract service. This results even in an additional modeling effort. Grouping of services which can be used in an analog form is the main benefit.

5.4.5 Transformation of a Business Process into an IT Process

The objective of the top-down approach in a SOA is to automate the process. A transformation of the business process into an IT solution prepares the automation and the service orchestration. The software 'SOA Architect' provides a BPEL model type that can represent the BPEL formalism visually and an automatic model transformation from EPC into BPEL. The figure 5.18 shows an extract of the transformation of the EPC depicted in the figure 5.8 into the BPEL formalism. Technical objects like 'scope begin', 'scope end' or 'assign' are omitted in the extract to better emphasize the interesting objects.

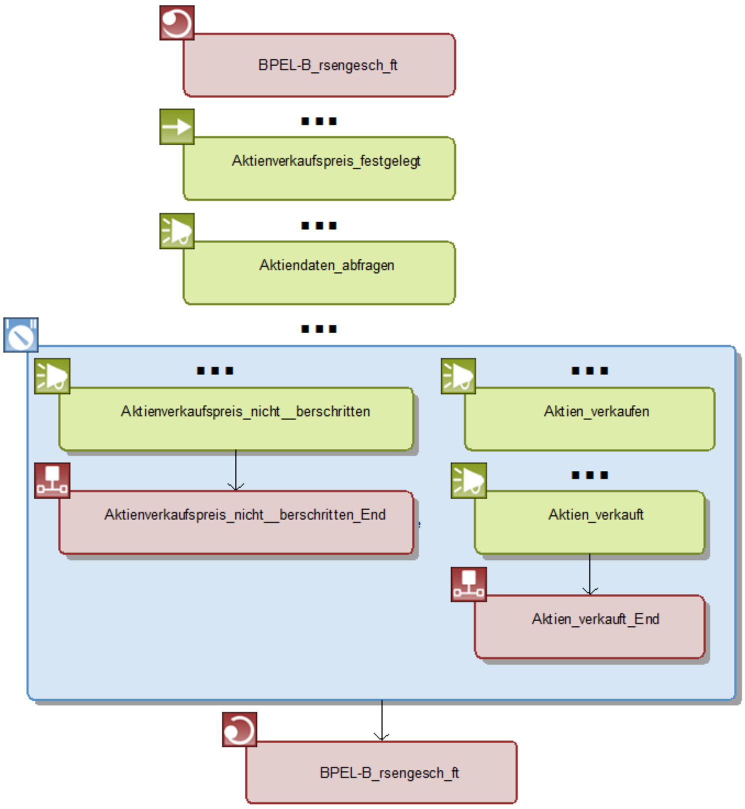


Figure 5.18: EPC model transformed into BPEL

The figure 5.19 shows the same process as the figure 5.8, but localized in English.

Like shown in these two examples, BPM software supports multi-language modeling. But software development environments support only mono-language developing. Software program code is only written one time in a predefined language. It

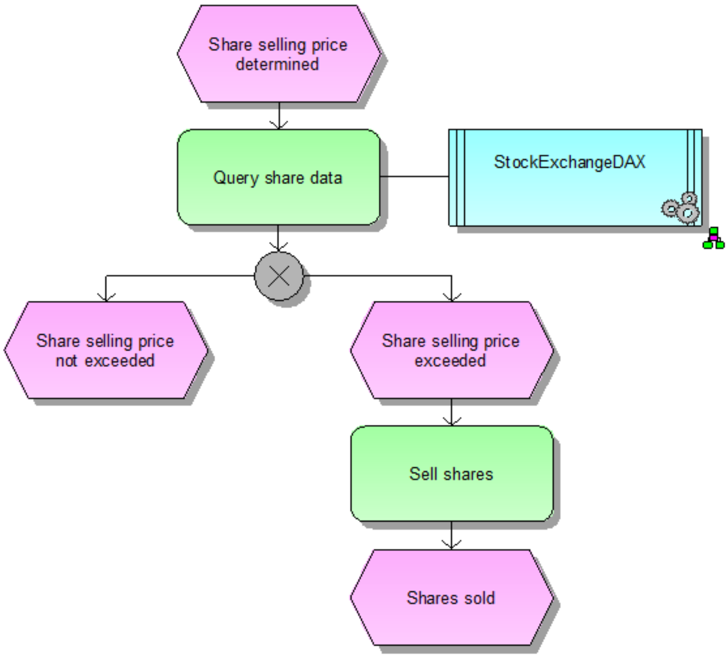


Figure 5.19: EPC translated in English

means that the variable and function names are never localized. In most cases, the development language is set to English, to facilitate off-shoring and code sharing between subsidiaries.

The figure 5.20 shows an analog extract of the BPEL process generated from the English EPC. This example outlines the major problems with the multi-language modeling concept. The transformation functionality uses the name of the events and functions of the source process as technical names in the BPEL process. In this example, at first German language is generated, then English language. In the German language, additional problems are caused by characters with umlauts, which are simply replaced by underscores.

To solve this behavior a simple solution is to maintain an attribute containing a technical name that will be used during the transformation process. This step requires the same effort than a process localization: the object names are localized in a technical language. The quality of the technical object name depends on the modeler and no standard denominations are set.

The generated BPEL process is not complete at this stage. It will be exported in the BPEL XML format and reimported in a development environment where the last required technical details are added. In a round-trip scenario, the supplemented BPEL process can be reimported in the modeling environment and changes at the business process level (EPC) can be merged with the reimported BPEL process. But if additional process steps were created in the development environment, the round-trip

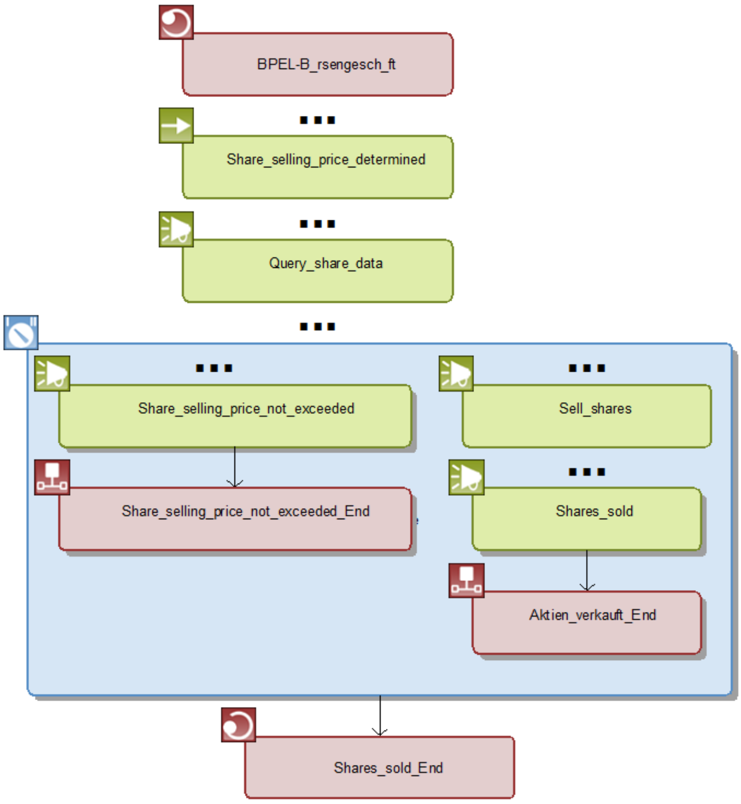


Figure 5.20: English EPC model transformed into BPEL

process cannot create business activities in the high-level process with an according name in natural language from the BPEL technical name.

5.5 The Process Execution Phase

In the current context, process execution means the manual execution of approved processes and not the execution of automated processes. Parts of the business process models are published in the corporate intranet for consulting purposes or for interacting with different human process participants. If a process participant wants to consult which steps he has to complete to execute a task, he must find out where the adequate process model is stored in the mentioned intranet.

The standard search functionalities of BPM tools offer the possibility to search by object name with regular expressions and by linked attributes like 'creation date'. Pattern matching is insufficient to retrieve all variants. Searching functionalities over the business repository are suffering from the object inconsistencies like described in section 5.3.2. When similar objects are stored in the database, variants may not be found. For example, searching for 'Kunde' will not find objects containing 'Auftraggeber'. The description attribute of a business object may contain useful information written in natural language. The same finding restrictions apply on description, definition or comment attributes. The business repository search is also an issue during modeling time because a process modeler may use the integrated search function to support their modeling tasks.

To favor the process retrieval, business process models may be tagged with key words. Analog to the business repository search, key words are natural language items, which can be inappropriate. The process participants may search with a synonymous term or a spelling variant and thus do not find the process model.

In a global company, process participants can be issued from different countries. If not all of the process models are localized, the entered search string can appear in a different language when participants assume to find models translated in their native languages.

A last issue concerns the linking of process documents discovered at design time. Documents may contain finer-grained information that is not available in the process models. Ideally, documents are linked to the process step object, a procedure that is sometimes done manually during modeling time. If not, an appropriate context search over a CMS or a process-specific navigation utility would be very appreciable.

5.6 The Process Optimization Phase

The core method of process optimization is to merge several existing process models and create a new process model that integrates the most efficient and needed activities from the existing processes. A possible use case is when two or more departments of a company are doing similar tasks and synergies are aimed. Another use case concerns company acquisitions. The processes of both companies must be merged to achieve their objectives in a more efficient way.

An example for this is the modeling of custom laws of a company with several subsidiaries in different countries: at regular intervals, the country specific databases are merged in a central database, in order to align their processes. The figure 5.21 shows the case before 2008, Germany and USA have created their own process models, reflecting their specific workflows. Until 2011, they aim to align their processes, to create an optimized process that encompasses all the process activities of the subsidiaries. Once the processes aligned, new process models are created. A few years later, this new process model will be merged with the processes of the French subsidiary and later on, with the Russian one.

To do that, single process steps must be found in both databases. Inconsistent and redundant business objects let the business repository consolidation become a difficult task. Analog process steps may not be detected and optimization is affected by that. In case of company acquisitions, both companies may have modeled their processes based on a different terminology. BPM tools provide multi-language modeling support.

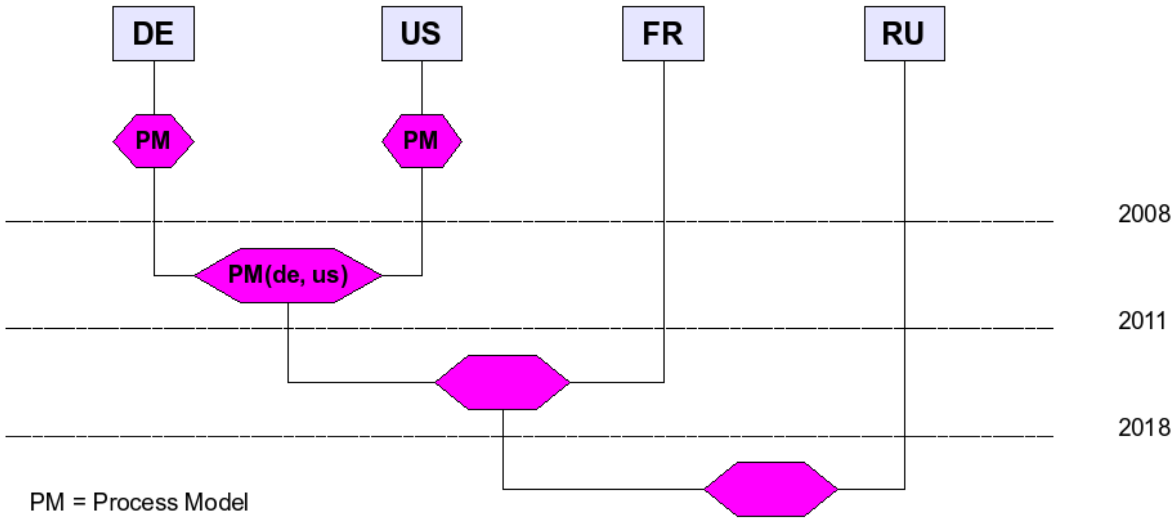


Figure 5.21: Merge example

If both subsidiaries have modeled their processes in their native language, it will be even more difficult to find equivalent business objects. If merging activities are suffering from the inconsistencies, the amount of inconsistencies can even increase due to merging activities.

5.7 Issues Summary and Quality Impact

Like described in previous sections, language issues at modeling and design time have consequences during the whole process management life cycle. Basic rules are available to develop a quality assurance model for business object labeling. This section develops the quality attributes and metrics required at different levels.

The overall objective of Business Process Management is to react rapidly to new business challenges and requirements of the market. The requirements on *business process execution* (manual or automated) are:

- *Agility*: all process participants have to identify at their levels which appropriate actions they have to perform to respond on any occurring situation.
- *Flexibility*: if a process workflow needs a change, the adaptation must rapidly be undertaken.
- *Interoperability*: for a better interoperability, all process participants, whether human or machine, should talk the same language and exchange the same data objects within one organization. Cross-company interoperability is not the objective in this concept.
- *Efficiency*: processes should be performed as fast as possible.

- *Optimization*: redundant process steps must be avoided and process results must be reused if possible.

Since business process modeling is the method to achieve the business process objectives, every business process requirement itself postulates prerequisites from business process models:

- *Findable*: a process must be easy to find to assure process *agility*.
- *Localized*: a localized business process model assures process *agility* and *interoperability* in a global company.
- *Mergeable*: if analog processes can easily be merged, it increases the business process *optimization* and *efficiency*.
- *Automatable*: if a process can easily be transformed in an IT solution, it increases the process *flexibility*.
- *Easy to analyze*: if a repository is easy analyzable, it supports the business process modeling tasks in particular during the design time by favoring impact analysis.

The business repository stores the objects modeled in business process models. To fulfill the quality requirements of business process models, the business repository must conform to following criteria:

- *Consistency*: a consistent business repository favors the process model *mergeability*, the *findability* and the *translatability* from a qualitative point of view.
- *Consolidation*: the business repository consolidation procedure searches for look-alike business objects. If the business repository contains only consistent objects, the consolidation can tap the full potential. A consolidated repository facilitates the *mergeability*, the *findability* and the *translatability* from a quantitative point of view.
- *Translatable for localization*: if a business repository is easily translatable, the localization process is cheaper and faster. As a consequence, process models using already translated items are promptly available in a *localized* form.
- *Translatable for IT implementation*: if a business repository can be translated in an IT specific language (programmatic interface or object names), it favors the process *automation*.

To favor a consistent, consolidated and translatable business repository, each business object must fulfill the following quality criteria:

- *Correctness*: correct spelling and grammar
- *Appropriateness*: an appropriate, accurate, exact term has to be used
- *Coherence*: the semantics of the business object has to be coherent

- *Readiness*: the business object must be readiness rules compliant

A company decides at process governance level which resources, methods and software support are ensuring the business repository consistency. Rules, terminologies (corporate or standard set of business vocabulary) and methods must fulfill the following quality criteria:

- *Exhaustiveness*: ideally, all terms occurring in business objects must be known.
- *Accuracy*: each term must have a strong and precise semantics.
- *Restrictiveness*: for one concept, only one denomination must be allowed. Synonyms are unwished. Compliance rules must define strong object structure patterns.
- *Sustainability*: terms must be sustainable to avoid legacy data and interoperability breakdowns.
- *Reliability*: the modeler must dispose of trustful software support to assure the compliance with the business object denominations quality metrics.

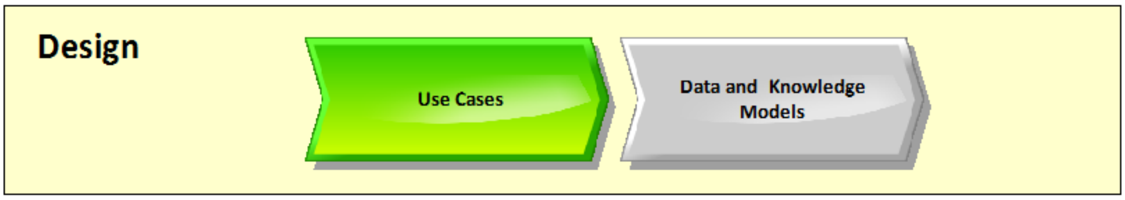
Business process related documents are supporting the process design and the process execution. Following criteria must be fulfilled to transform local information units like documents stored in a CMS in a process-driven information flow that is fully exploited at design time and runtime.

- *Findable*: documents may contain additional information that must be findable to increase process *agility*.
- *Linkable*: documents can be linked to process models to favor a process-driven consulting.
- *Classified*: a process-driven classification system can be placed on top of the document structure to provide a navigation that is intuitive to the process participants. It will increase the process document findability and in turn the process *agility*.
- *Interpretable*: at design time, if parts of documents could be interpreted, it would support the modeling tasks by generating automatically appropriate business objects.
- *Aligned*: ideally, documents and business process models are aligned to ensure up-to-date information and to find out the right decision.

As a conclusion, standards and best practices enable interoperability of data, which, in turn, maximizes the potential for access to information, ensures longevity and usability of data, and improves the efficiency of processes for producing, localizing, distributing and monitoring information and information flows.

Chapter 6

Life cycle Support and Optimization through Semantic Technologies



The quality of business object denominations impacts the whole process life cycle like described in the previous chapter. This chapter describes the functional requirements at each life cycle stage and demonstrates the benefits of applying semantic technologies to business process modeling and business process management related tasks. The objective is to achieve a continuously process improvement and a better alignment between business an IT processes. The functionalities are depicted from the user, system owner and governance point of view with help of user interface mock-ups. The mock-ups are enriched with samples which demonstrate a panel of difficulties inherent to natural languages.

6.1 Overview

This section gives a short summary of the actors, functions and features which are used in the further use cases description.

6.1.1 Actors

Process modeler: process modelers are experts in representing business processes with the help of different modeling notations and conventions.

Technical process modeler: whereas process modelers are modeling the processes from a business point of view, the technical modelers concentrate on the technical requirements necessary to the future process execution.

Process owner: the process owner is in charge of the process design, quality, execution and optimization.

Process participant: process participants are the people who are executing the processes. The participants are either employees of the company or external supporters.

Business content expert: business content experts offer specialized knowledge about modeling naming conventions, including business expressions structure, corporate rules and terminology. Ideally, it comes with terminology expert knowledge of the company industry sector.

6.1.2 Functions and Features

Business process modeling environment: the business process modeling environment is a means to represent business processes visually.

Business process collaboration platform: business collaboration platforms offer a set of software components and software services which enable process modelers to model their processes or to upload their process models and allow process participants to find the information they need to execute the processes they are involved in. All participants are able to communicate and work together to achieve common business goals.

Business knowledge base: the business knowledge bases are a set of business specific terminology represented as an ontology. The knowledge bases are the prerequisites of the system. The following use cases assume that preset knowledge bases are already available, including a standard knowledge base and domain-specific ones. How the standard knowledge base is built will be described in the chapter 7.

Corporate business content collaboration platform: the business content collaboration platform offers a set of functions which enable the business content expert to manage the knowledge bases necessary to implement the different use cases. This user interface is hosted by the company.

Community business content collaboration platform: the business content collaboration platform offers a set of functions which enable the business content expert to manage the standard knowledge bases.

Corporate business content server: the corporate business content server is a server hosting the knowledge bases and the corresponding applications. The hosted knowledge bases can be extended and improved in a collaborative way, but are only company internal.

Community business content server: the community business content server is a server hosting the standard and some domain-specific knowledge bases. Companies have the possibility to share parts of the knowledge base to delegate and simplify the administration. Goal is that the community knowledge bases cover enough terminology so that a company only has to register its very specific terms.

Services: in the current context, a service is a software component which can easily be integrated in a Service Oriented Architecture.

6.1.3 Interactions between Actors and Features

The figure 6.1 shows an overview of the main components and the interactions between the actors. The represented components are (red bullets):

1. The business process collaboration platform
2. The business process modeling environment
3. The corporate business content collaboration platform
4. The community business content collaboration platform
5. The corporate business content server
6. The community business content server

In the figure 6.1, business modelers, process owners and process participants (marked blue), the corporate business content expert (red) and the community business expert (green) are interacting on the given process.

- Relation 1 and 2: the process modelers, process owners and process participants interact with the business process collaboration platform (1) and with the business process modeling environment (2). Their activities consist in modeling, merging and searching tasks.
- Relation 4: the business process modeling environment as well as the business process collaboration platform communicate with the corporate business content server to support the modeling, merging and searching activities.
- Relation 5: data delivery from the server.
- Relation 3: process modelers and process owners interact with the corporate business collaboration platform. It consists for instance of requests for business terms.
- Relation 7: requests from process modelers and process owners are sent to the corporate business server. Knowledge base management activities are stored on the server.

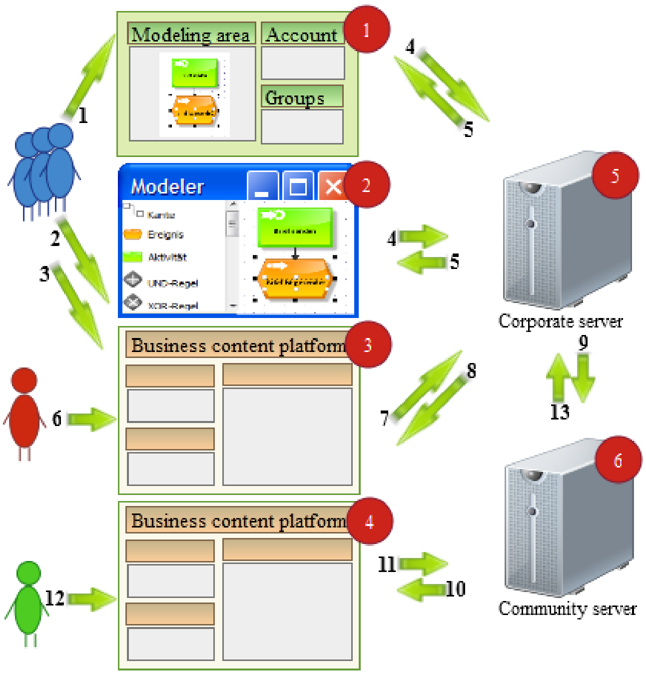


Figure 6.1: Interactions overview

- Relation 8: the requests must be validated by the corporate business content expert.
- relation 6: depicts the corporate business content expert activities: validating and managing the knowledge bases, as well as sharing some information with the community.
- Relation 9: the corporate business content server shares the information with the community business content server.
- Relation 10: the community business content platform displays the information to share for validation by the community content expert.
- Relation 12: the community content expert validates the shared information and manages the standard knowledge bases.
- Relation 11: the platform sends the information to the server.
- Relation 13: the shared information, which are now part of the standard knowledge bases, are sent back to the corporate server.

The description of the interactions will be deepened in the following use cases descriptions.

6.2 Use Cases

The use cases concern almost all phases of the process life cycle, excepting process controlling. They respond to the problems described in section 5. The figure 6.2 gives an overview of the use case distribution over the process life cycle and the following subsections provide a detailed description of each use case.

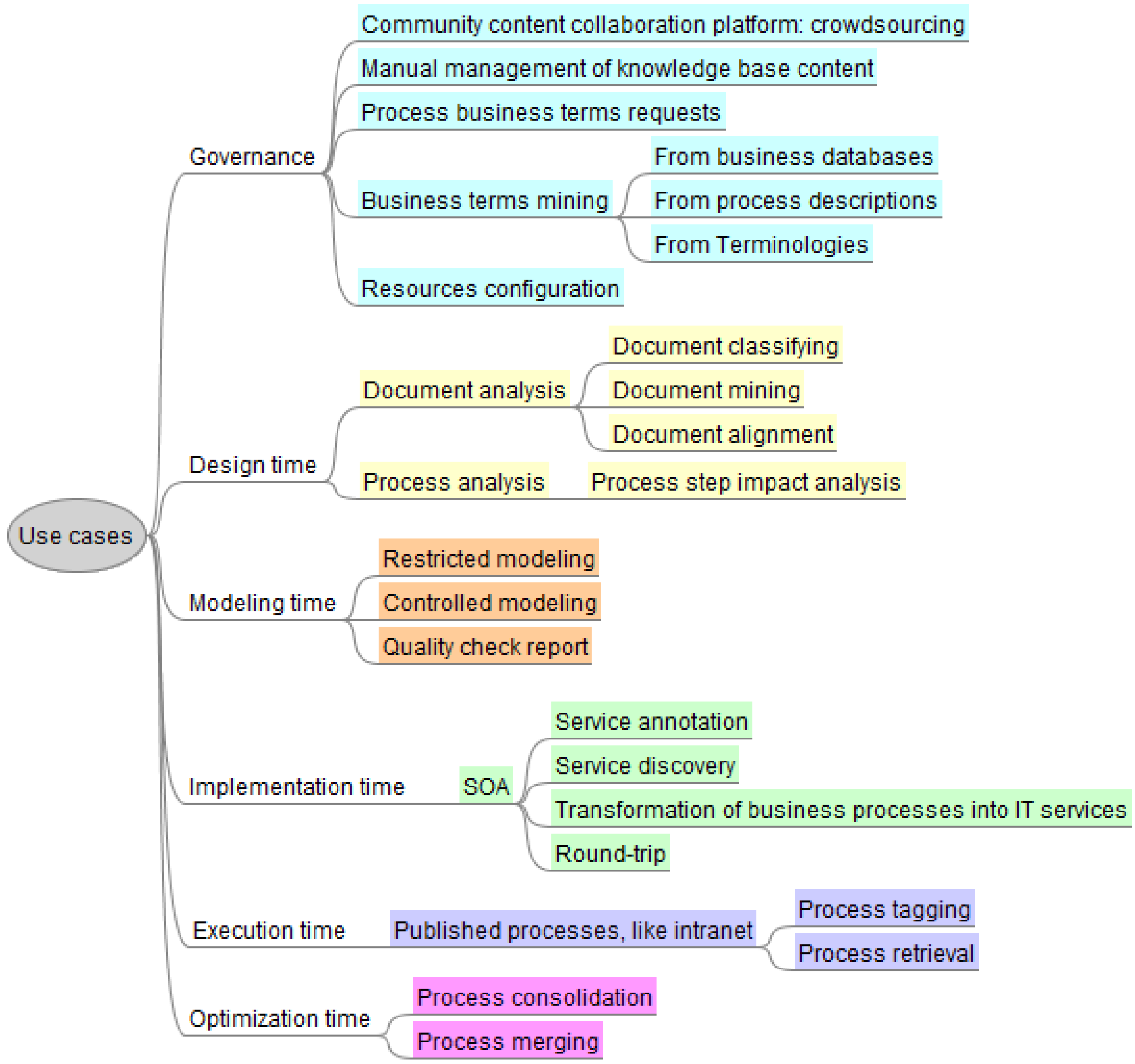


Figure 6.2: Use cases overview

6.2.1 Process Modeling Time

6.2.1.1 Restricted Modeling

The objective during the modeling time is to increase the quality of the business objects directly at the source in terms of correctness, appropriateness and readiness, to avoid time-consuming corrections afterwards. As soon as a business object is created in the modeling area, a set of quality assurance rules is applied that checks the compliance of the object denomination with given naming constraints. The rules are assessing a correct spelling, an appropriate usage of the terminology and the adherence to naming conventions. If the object name does not fit the quality assurance rules, appropriate error messages are displayed which invite the business modeler to reformulate. If possible, a naming proposition is displayed in order to automatically replace the erroneous denomination. One of the challenges is to generate correct propositions.

This approach, called 'restricted modeling', allows free naming of business objects, but restricts the possible input. The restricted modeling is based on preset knowledge bases and default readiness rules which lead to low management and maintenance effort for the organizations. The quality assurance rules that have to be implemented are issued on one hand from corporate naming conventions and modeling training material and on the other hand from the corpus analysis detailed in the next chapter. The restricted modeling helps reducing the inconsistencies and redundancies in the business objects repository, but the highest quality level cannot be achieved. This approach suffers from a lack of support when terminology which is not part of the preset knowledge bases is involved.

The restricted modeling use case is illustrated by screen mock-ups. The modeling environment ARIS express is located in the background. The screens in the foreground are part of the design and may be integrated in any modeling environments. The screens are displayed if an error is detected after the business object is created and named.

The figure 6.3 demonstrates the matching of a readiness rule for the event object 'Brief ist gesendet'.

The figure 6.4 shows an inappropriate terminology usage.

> 'Abnehmer' and 'Kunde' are declared as synonyms and the preferred form is 'Kunde'.
>
> The used activity 'kontaktieren' is declared as unspecific. The activity 'kontaktieren' may be replaced by 'anrufen'. If the activity 'senden' is preferred, it must be qualified with an object like 'Brief', 'Email' or 'Fax'. In this case, the proposition would be 'Email an Kunden senden', which has the structure '`Object` *an* `Agent` *senden*'. The agent 'Kunde' is inflected (Kunde-n).

The figure 6.5 shows again an example of inappropriate terminology usage.

> 'Daten' is a very unspecific term that can stand for many kind of data objects. This term must be specified in more detail to avoid ambiguities and object redundancies. All 'Daten' available in the knowledge base are proposed, alphabetically sorted. 'Kundendaten' is divided in two more specific terms, 'Neukundendaten' and 'Stammkundendaten'.
>
> A context filter involving process maps can be applied to reduce the propositions. The filter will be described in more detail in the next use case.

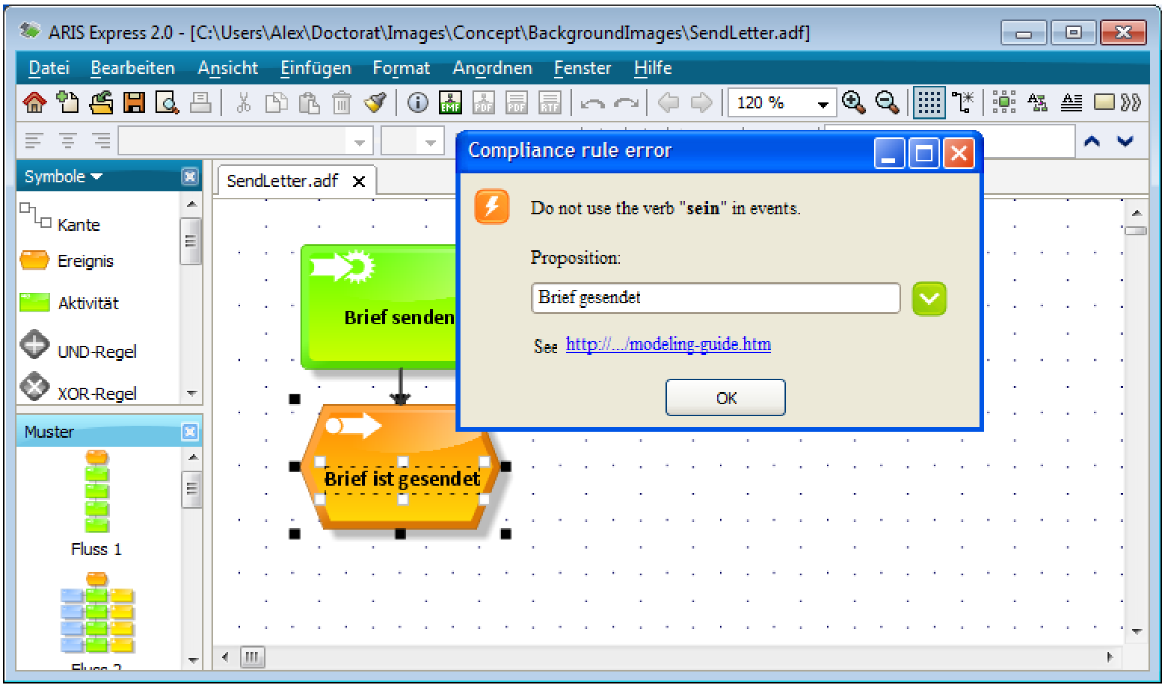


Figure 6.3: Restricted modeling example: structure check

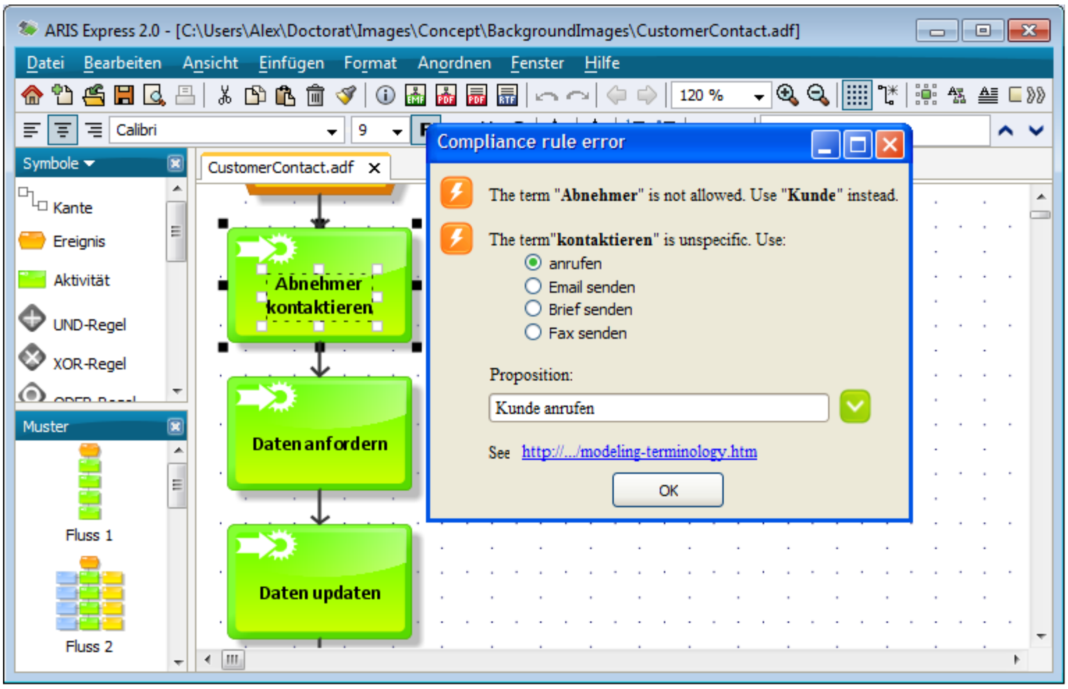


Figure 6.4: Restricted modeling example: terminology usage

The figure 6.6 shows a spelling error case and the same unspecific term usage as already shown in the previous figure.

The unknown word concerns the activity. A fuzzy match that picks the best

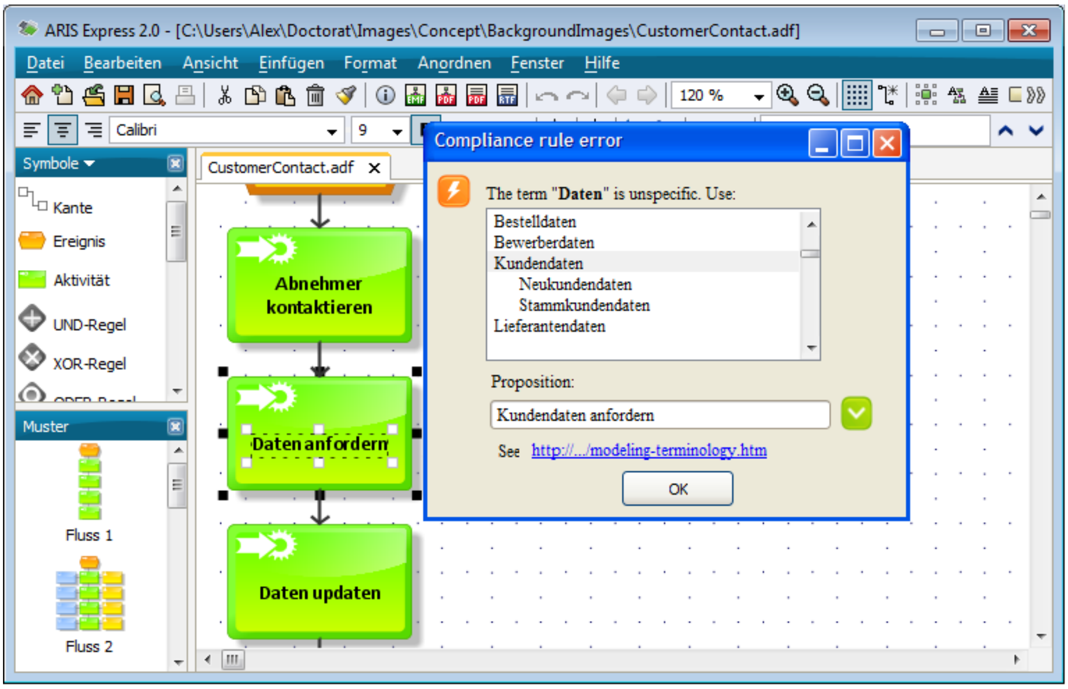


Figure 6.5: Restricted modeling: unspecific term usage

proposition between the activities associated to the object 'Kundendaten' must be implemented.

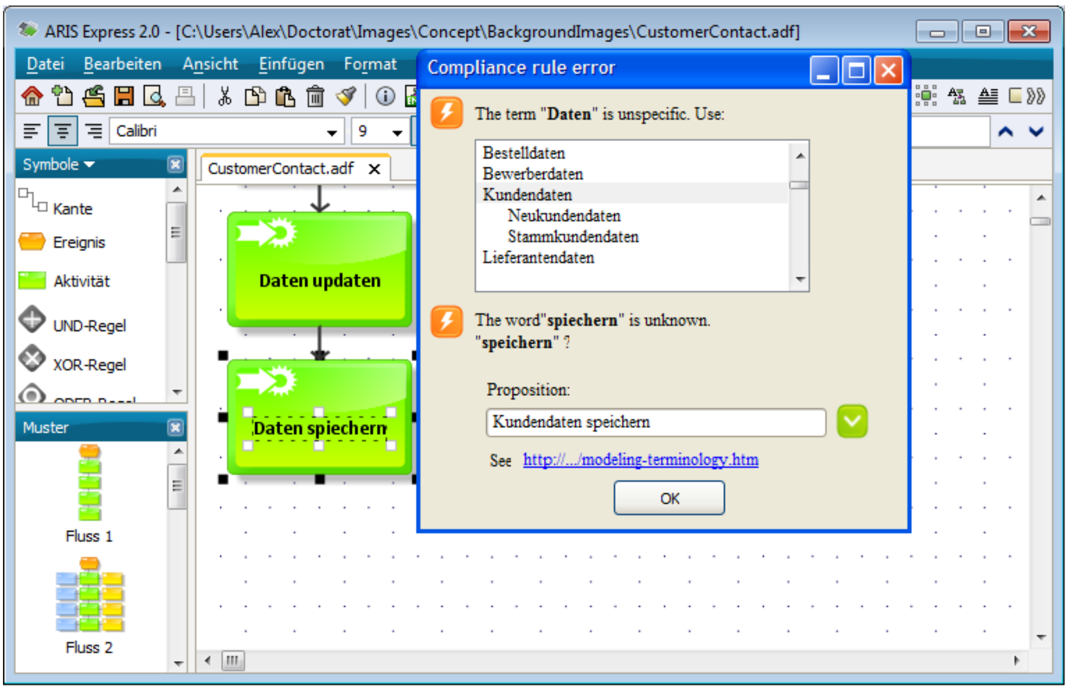


Figure 6.6: Restricted modeling: spelling error

The figure 6.7 shows a terminology usage warning for an activity.

> The term 'updaten' is deprecated and the synonym 'aktualisieren' must be used instead.

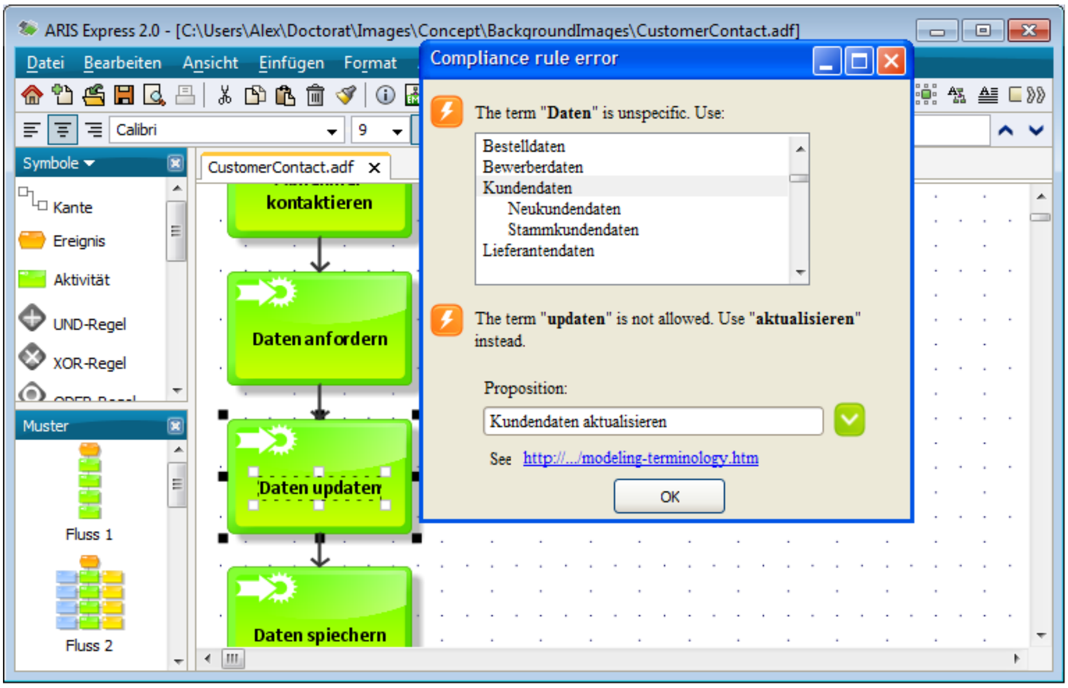


Figure 6.7: Restricted modeling: terminology usage for activities

The figure 6.8 shows the process corrected according to the restricted modeling application warning messages.

6.2.1.2 Controlled Modeling

The restricted modeling allows free object naming and subsequent corrections if errors are detected. The objective of the controlled modeling use case is to prevent the business modeler from entering erroneous or unknown object names. The advantages are that no corrections are needed after the objects are created and that a higher quality level can be achieved. The drawbacks reside in more administration efforts because all terms which are not part of the preset knowledge bases must be registered by the business content expert. In this case, the challenge is to provide functions and methods to simplify the administration tasks. Another negative argument could be the more time-consuming modeling process. Instead of directly creating the objects and typing the names in the modeling area, the process modeler must employ some wizards which will generate the needed business objects. The following figures show such a modeling scenario from the process modeler point of view.

The figure 6.9 shows the first wizard screen displayed when a new process model is created:

> A list of business terms is displayed in a list box. Business terms constitute the base of business object denominations. They must be selected (into the empty

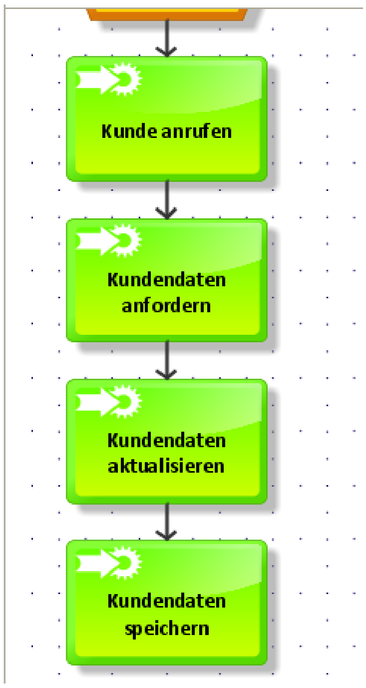


Figure 6.8: Restricted modeling: corrected process

list box) to access the next wizard steps. Goal is to compensate the time spent in using the wizard by creating several objects at once. Due to the amount of registered terms, navigation facilities must be provided, allowing a fast retrieval of needed terms. Three of them are represented on this mock-up.

The first one consists in a statistic cloud, displaying the most frequently used simple terms in business models. The values are a merge between values provided by the preset knowledge bases and statistics issued from the modeled processes.

The second one consists in a term classification involving a specialization relation between terms. The terms displayed in the list box at this stage are only simple terms like in the cloud. The list box comes with a search field.

The third navigation facility consists in an aspect oriented modeling feature allowing the thematic filtering of terms through a process map navigation. When the top-level processes are displayed, the list box is filled with all terms available in the knowledge bases.

The three top-level processes 'Management processes' (Führungsprozesse) 'Core processes' (Kernprozesse) and 'Supporting processes' (Unterstützende Prozesse) are always the entry points of the process map. Some elements of the process map should always be available like the top-level processes and some classical sub-processes categories. However, the process map can differ from one company to another, so that no static classification can be used. One topic in this dissertation will be the automatic classification of terms according to the process map.

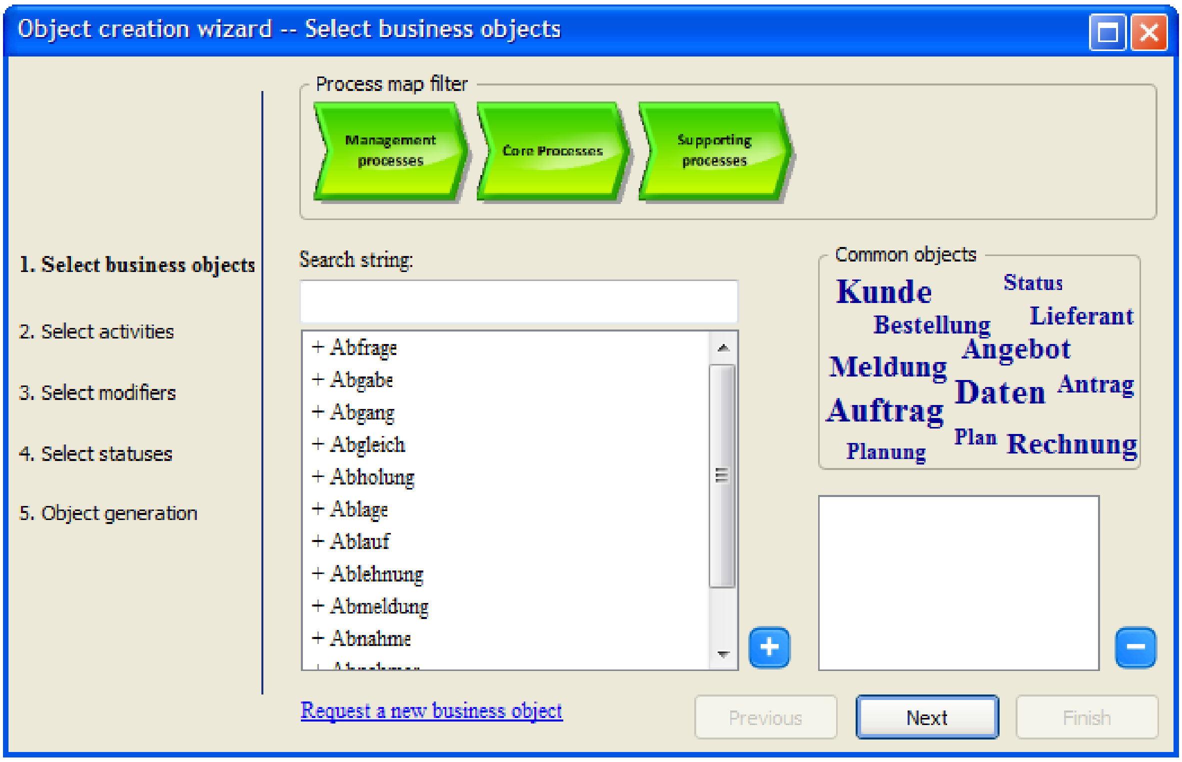


Figure 6.9: Controlled modeling: modeling entry point

The figure 6.10 displays the specialization relations of the term 'Daten':

> 'Daten' was searched in the search field. The node 'Daten' is expanded in the list box and all specific data terms are displayed in alphabetical order as subnodes. 'Abgleichsdaten', 'Abrechnungsdaten' are for instance a special type of 'Daten'. This example displays only compound terms but terms like 'Daten der Untersuchung' are also considered as a specialization of 'Daten'. The goal is to automatically classify the terms by specialization.
>
> As soon as the node is expanded, the cloud is updated with the most used 'Daten' terms.

The figure 6.11 displays a first navigation level of the process map from 'Supporting processes':

> A set of supporting processes is displayed and 'Human resources' (Personalwesen) is selected.
>
> The terms are filtered accordingly in the list box and in the cloud. Simple terms commonly used in human resources related processes and subprocesses are displayed. The simple nodes can further be expanded to display the specializations.

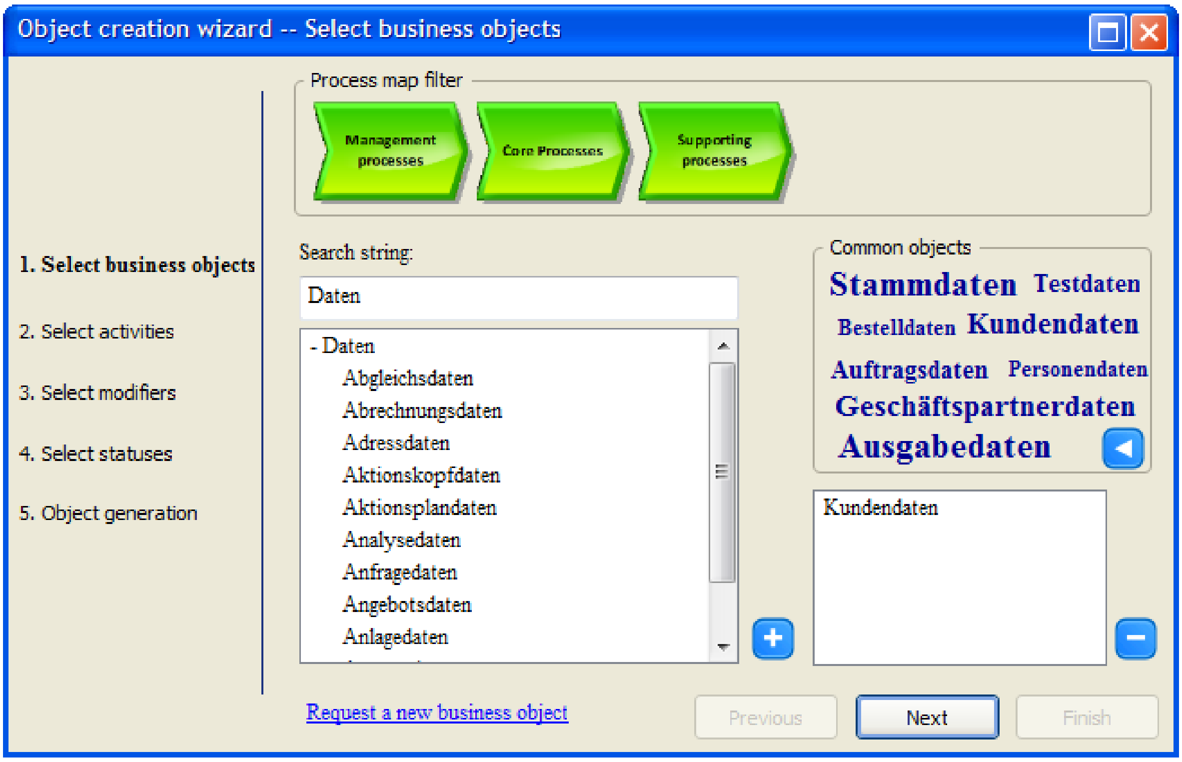


Figure 6.10: Controlled modeling: specialization filtering

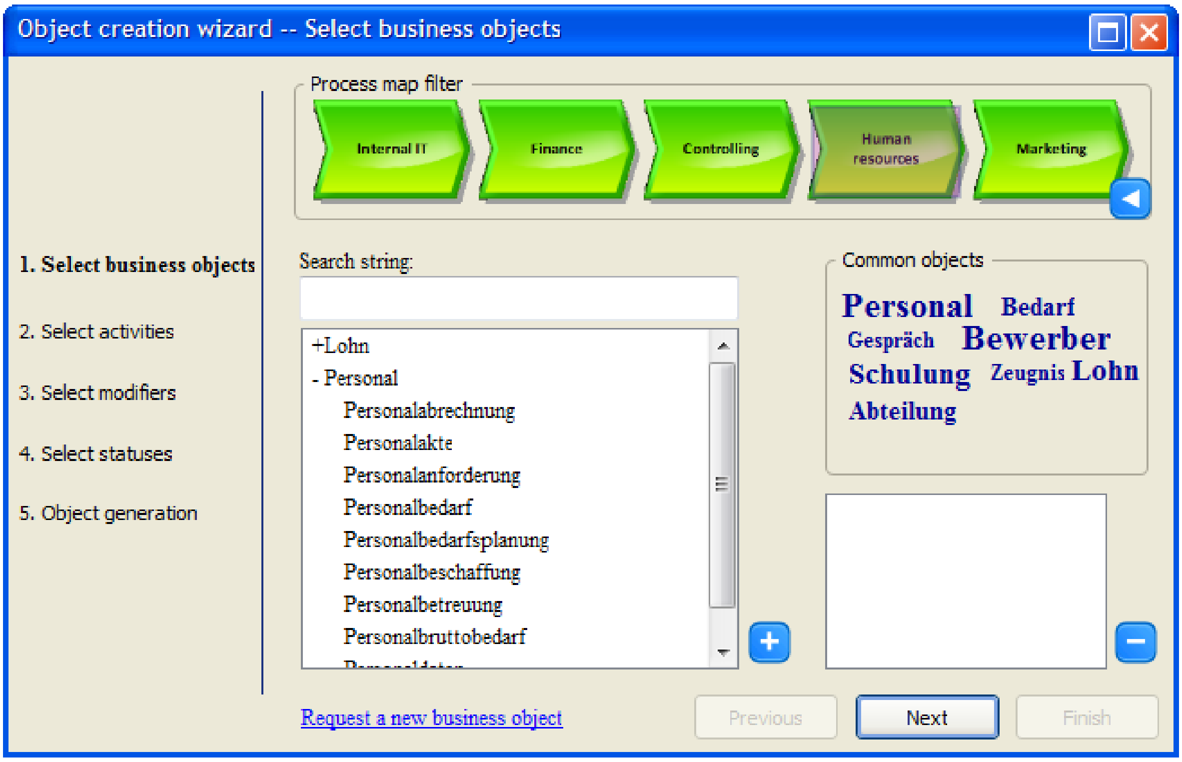


Figure 6.11: Controlled modeling: aspect oriented filtering

The figure 6.12 shows a lower navigation level of the process map from the node 'Human resources':

> Three sub-process categories are displayed and the 'Personal procurement' (Personalbeschaffung) node is selected. Again, the terms are further filtered according to the selected category.
>
> The search text field not only allows to find simple string patterns. Available synonyms are found as well if the search pattern is a regular term.
>
> If some desired terms are not available at this point, the process modeler has to request a new term by following the link 'Request a new object' in the wizard. He will be redirected to a specific page in the business collaboration platform, which will be detailed later.
>
> The screen shows that the process modeler has selected the term 'Bewerber'. For the next wizard screens, we assume that he also selected 'Bewerbungsunterlage' and 'Personalbedarf'.

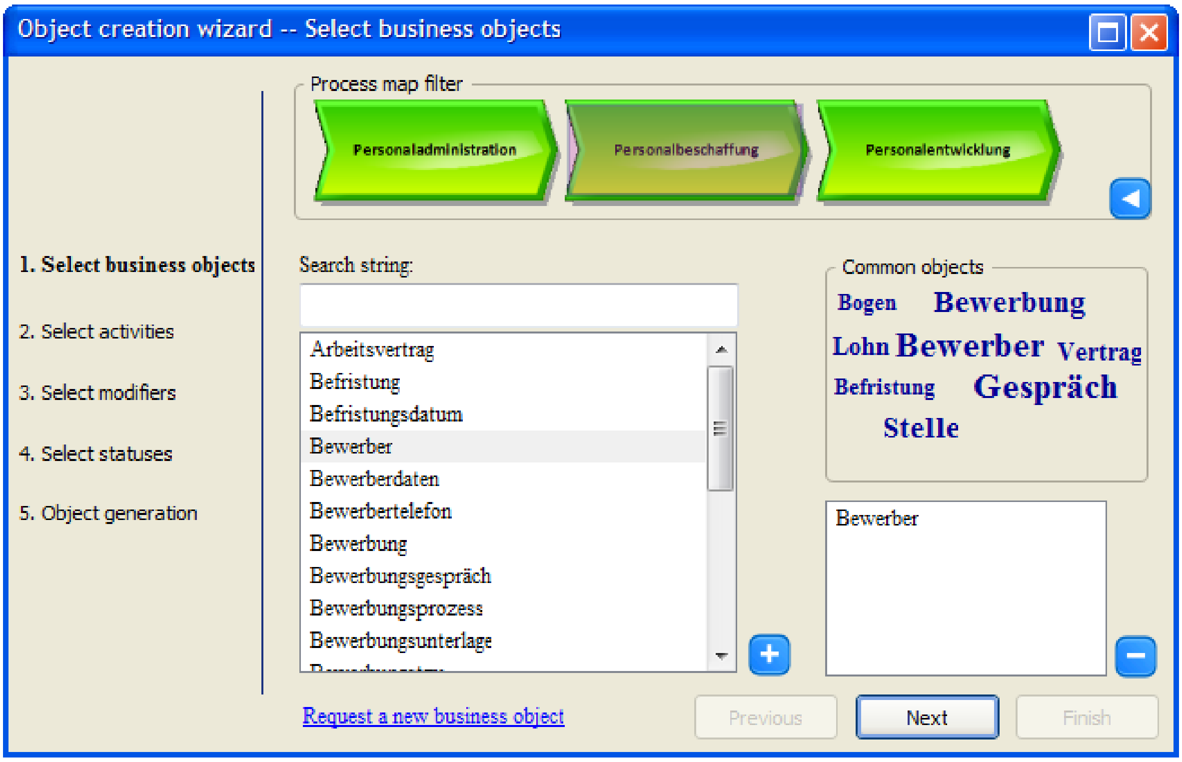


Figure 6.12: Controlled modeling: aspect oriented filtering and selection

At this point, the business terms involved in the process are selected and the process modeler can switch to the second wizard step, to select the activities corresponding to the business terms. The figure 6.13 shows the second wizard step:

Each selected business term is displayed in its own register card. The register cards are marked with a green or orange flag. The green flag means that at least one activity is now selected. The orange flag means that the term was not already processed. This step allows the creation of process steps.

If the desired activity is not available in the list, the process modeler has to click on the link 'Request a new activity' which leads to a request page in the content collaboration platform.

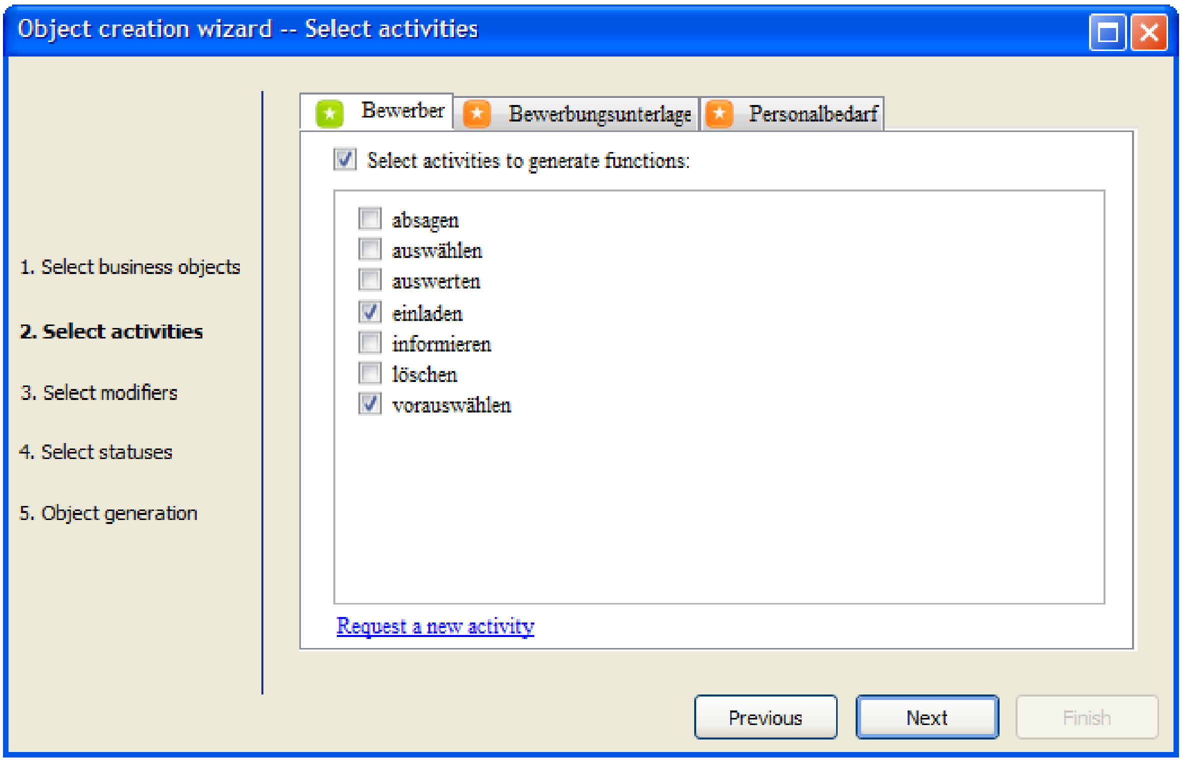


Figure 6.13: Controlled modeling: select activities for 'Bewerber'

The process modeler must know in advance the rough design of its process to reach an efficient modeling. This counts especially for the business terms. Concerning the activities, the fact that the wizard proposes a set of associated activities can help him to remind some forgotten process steps or sub-steps, which compensates again the wizard usage effort.

The figure 6.14 is analog to the previous figure, but the register card 'Bewerbungsunterlage' is selected.

In the figure 6.15 the register card 'Personalbedarf' is selected:

In this case, no process step involving this term is desired. The checkbox 'Select activities to generate functions' is then deselected so that the register card status can be switched to green.

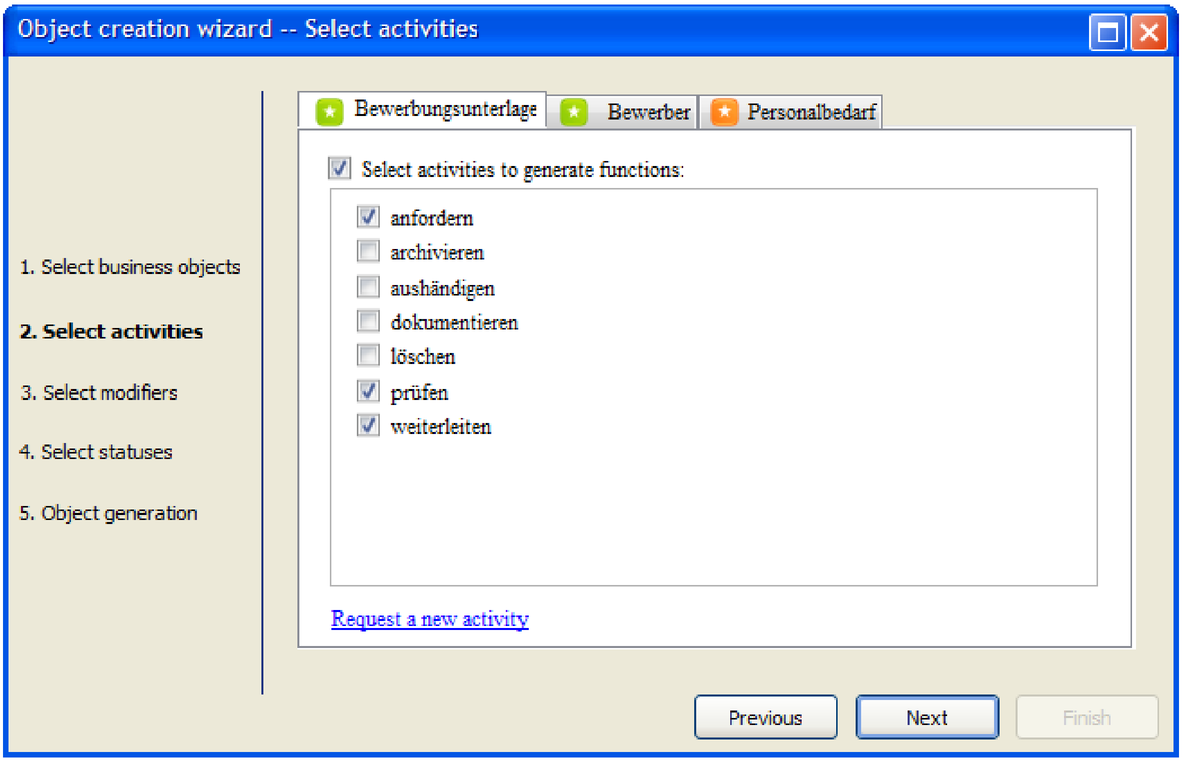


Figure 6.14: Controlled modeling: select activities for 'Bewerbungsunterlage'

At this point, the next button is enabled to go to the next wizard step.

The next wizard stage concerns the generation of process step objects. At this point, the names of a process step are composed of a business term and an activity name. Some object names can be specified in more detail, for instance with object and activity modifiers. The figure 6.16 shows some examples of modifiers completion:

> A 'Bewerber' can be 'intern', 'extern' or not specified like in the second line of the table. The activity 'prüfen' can be 'automatisch' or 'manuell'. In this case, it is to be asked whether 'automatically' or 'manually' better would not be represented by different function types like a manual or automated task. The activity modifier would then be superfluous. Only standard function structures are represented in this screen mock-up. Some business object names are taking different arguments. For example, the activity 'senden' can have an agent to which the object has to be sent as an optional argument. The activity 'anzeigen' associated to the object 'Meldung' can take a literal as an optional argument, specifying the string to be displayed.
>
> If the process modeler misses a modifier, he can request one by selecting the concerned line and activating the link 'Request a new modifier' or 'Request a new activity modifier'.
>
> Once the potential modifiers are selected, the process modeler can switch to the next wizard step.

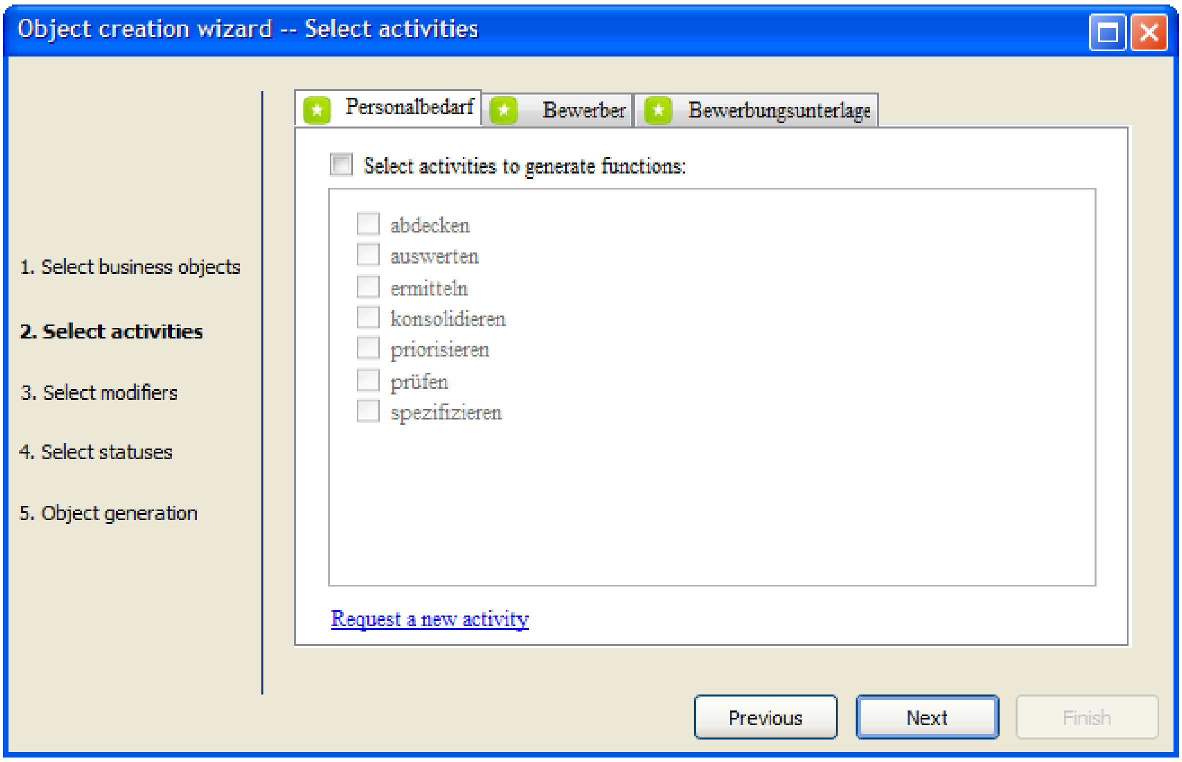


Figure 6.15: Controlled modeling: no activity selected for 'Personalbedarf'

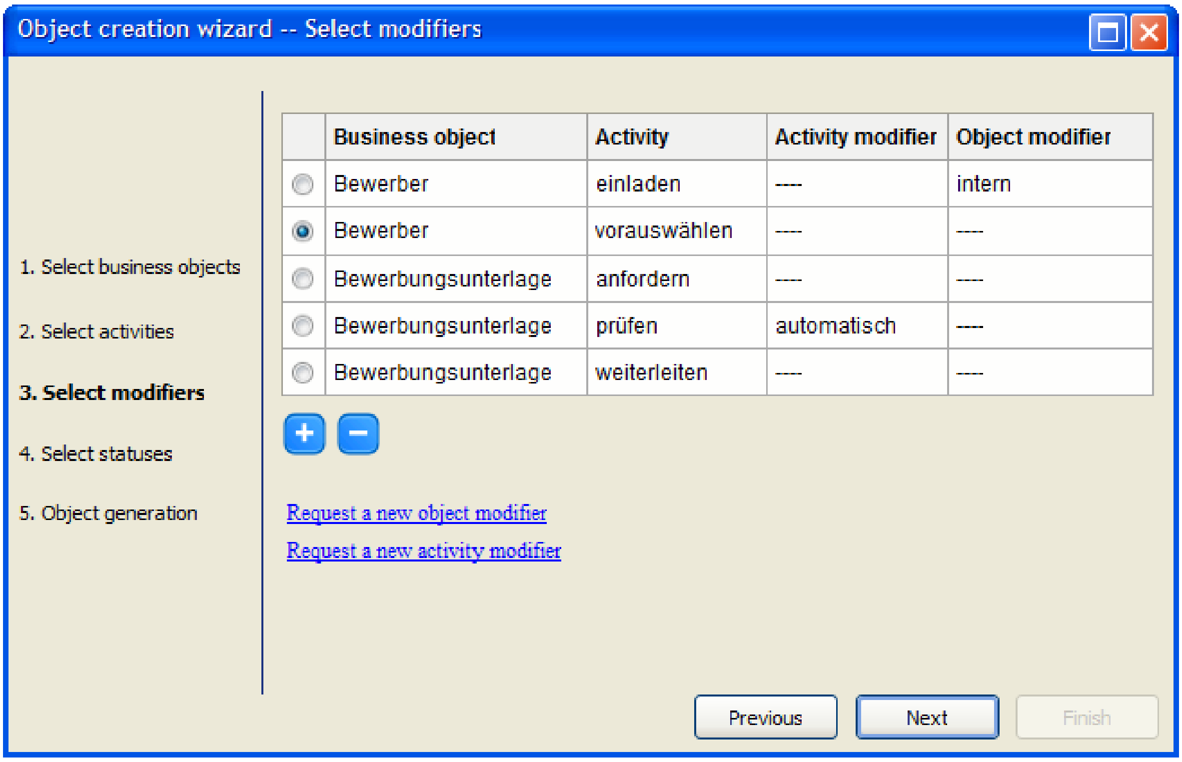


Figure 6.16: Controlled modeling: select modifiers

The next step concerns the generation of event objects. The figure 6.17 shows the statuses available for the term 'Bewerber':

> Again, each business term is placed in a register card. A list of statuses is displayed for each term. The green and orange flag system is analog to the second wizard step.
>
> Statuses corresponding to previously selected activities are emphasized in group boxes.
>
> Statuses corresponding to activities which were not selected before are placed under the group boxes.
>
> Finally, a list of adverbs is visible at the right side.
>
> If no event generation is wished for the business term, the check box 'Select statuses to generate events' must be deselected.
>
> It is noteworthy that the statuses appear in the positive and negative form. Two standard ways for expressing negative events exist: using the 'nicht' particle with the positive status (not possible = nicht möglich) or the opposite term if available (impossible = unmöglich). Sometimes the particle 'kein' is used. To standardize the usage of negative forms, the form including the 'nicht' particle is favored.
>
> If the process modeler misses a status, he can request one by activating the link 'Request a new status'.

The figure 6.18 shows a list of statuses available for the term 'Personalbedarf':

> Due to the fact that no activity was selected for this term, no status is emphasized.

The process modeler can now switch to the last wizard step. The last step consists in a summary of the process steps (functions) and events to be generated. The figure 6.19 shows the function list constituted with the elements selected in the previous wizard step:

> The process modeler can delete unwished objects in this step. It is noteworthy that the term 'Bewerbungsunterlage' is in the plural form.

The figure 6.20 shows an analog screen for the generated event objects.

> The wizard can now be completed by clicking on the finish button.

The wizard result consists in a generated process model (here epc), like shown in the figure 6.21:

> The objects listed in the last wizard step are generated.
>
> The objects can now be placed like needed.

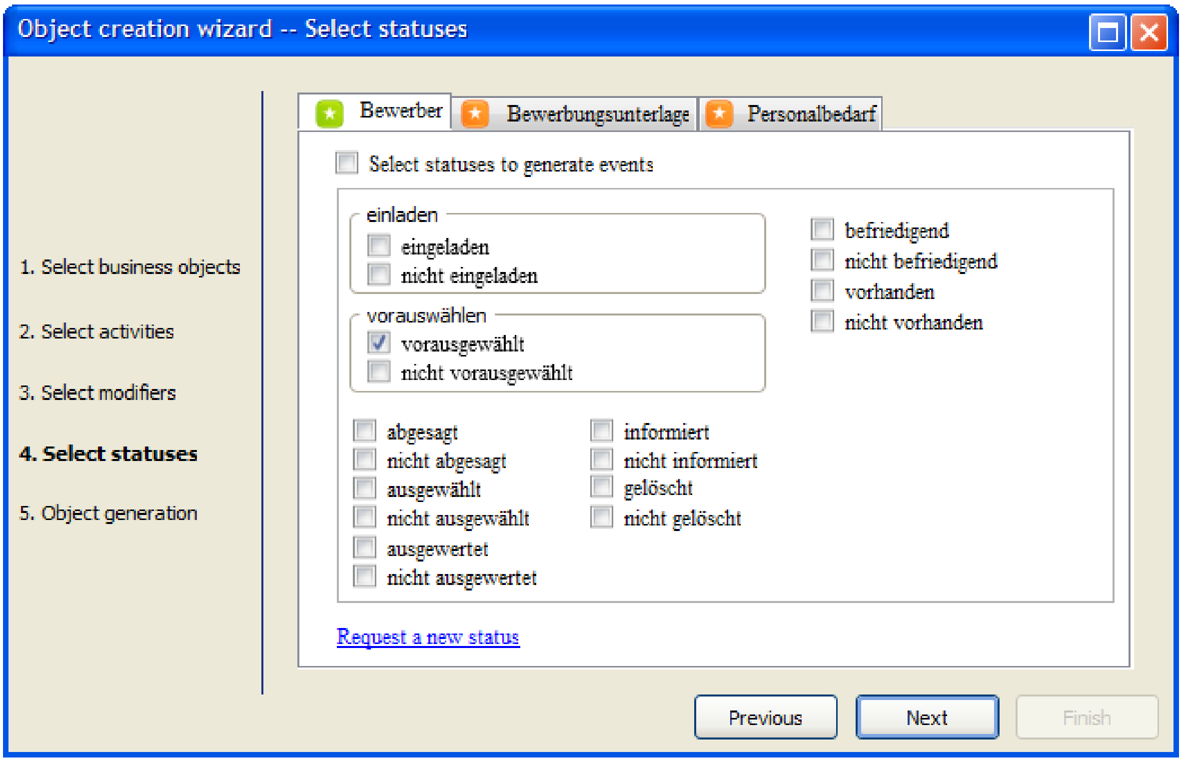


Figure 6.17: Controlled modeling: select statuses for 'Bewerber'

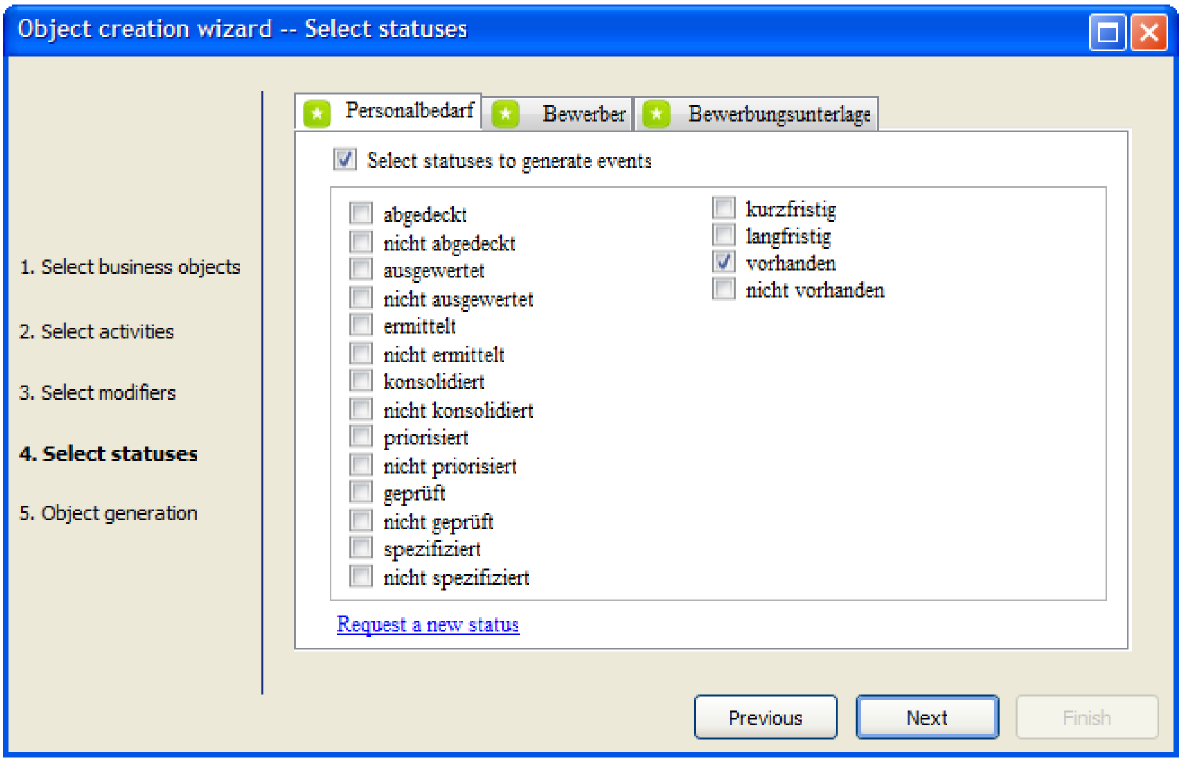


Figure 6.18: Controlled modeling: select statuses for 'Personalbedarf'

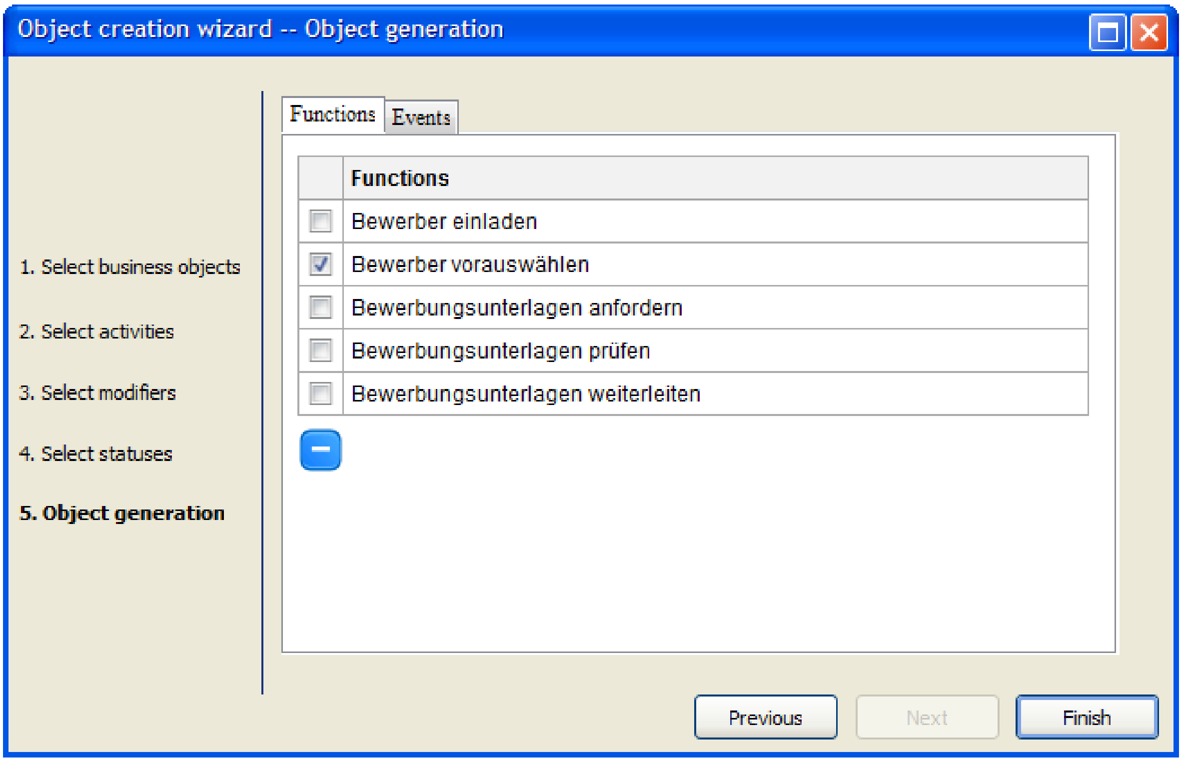


Figure 6.19: Controlled modeling: summary of functions to be generated

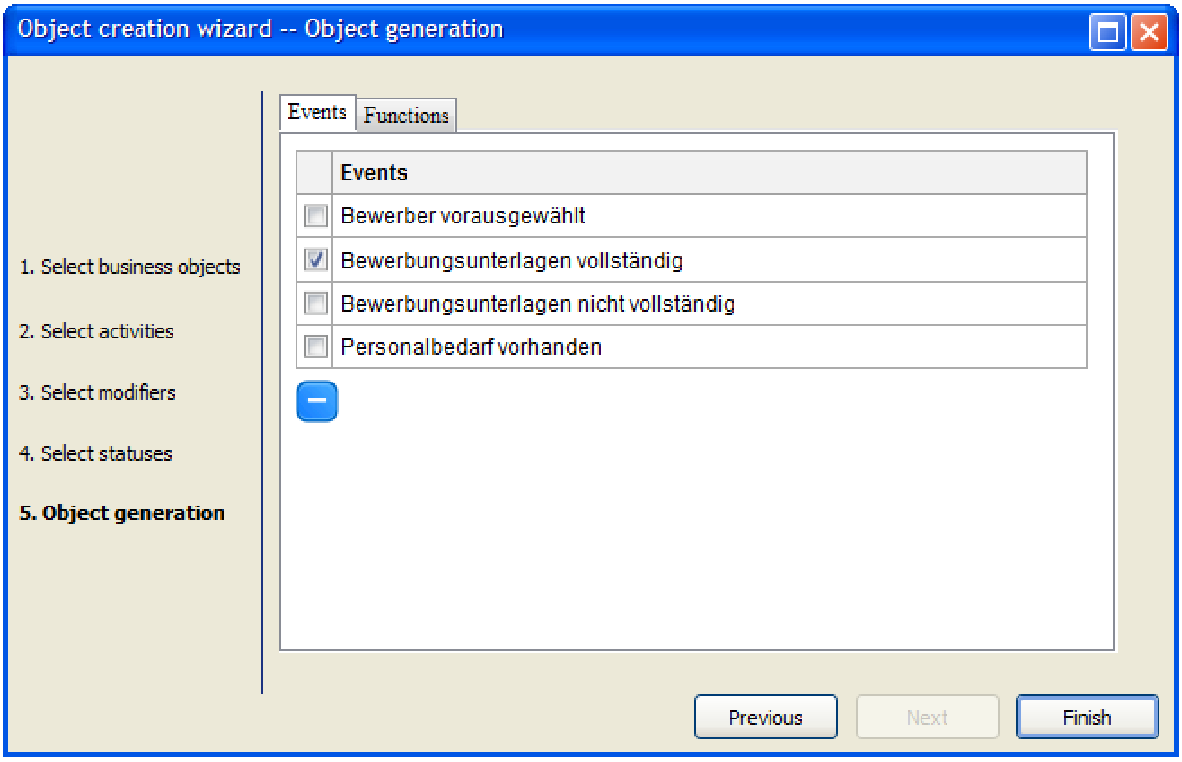


Figure 6.20: Controlled modeling: summary of events to be generated

The event 'Personalbedarf vorhanden' is placed as the first element because it could be a start event: no process step including the term 'Personalbedarf' is available.

Some relations are also generated. 'Bewerber vorausgewählt' is probably the result of the process step 'Bewerber vorauswählen'. The events 'Bewerberbungsunterlagen vollständig' and 'Bewerbungsunterlagen nicht vollständig' are mutually exclusive so that they are joined by an 'xor' connector object.

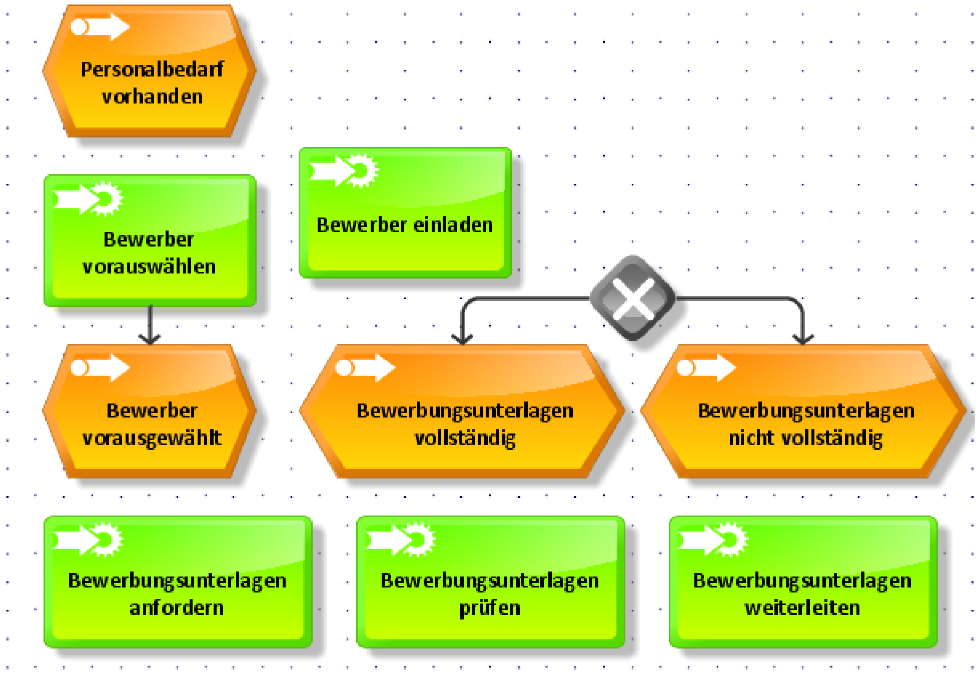


Figure 6.21: Controlled modeling: generated objects in an epc model

Once the model was generated, the process modeler may have forgotten some business objects. Functionality has to be provided to avoid, when possible, the usage of the wizard to increase the process modeling efficiency.

The figure 6.22 shows the creation on the fly of a new process step:

Instead of allowing the free naming of the objects, a menu is displayed. The main menu displays the objects previously selected in the wizard, because the probability is high that the process modeler wants to model a further function concerning the same business terms.

If one or more other objects are wished, the wizard is again opened to search for them by selecting the menu point 'Other objects...'. In this example, the term 'Bewerbungsunterlage' is selected. In a sub-menu, all activities associated to the term 'Bewerbungsunterlage' are displayed excepting those already generated.

If modifiers are needed, the wizard will be opened to the corresponding page by clicking the 'Open wizard...' sub-menu item. In this case, the process modeler created the process step 'Bewerbungsunterlagen löschen, which again is in the plural form.

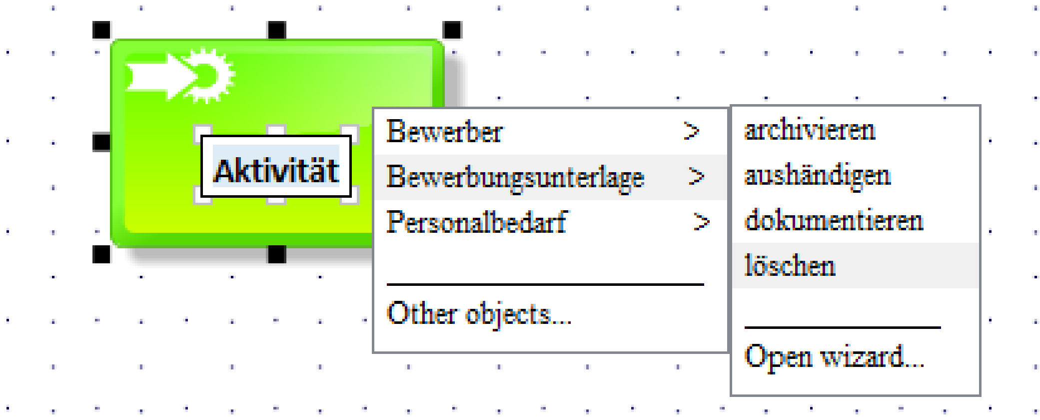


Figure 6.22: Controlled modeling: creating a new function

After the process step object is created, the process modeler wants to associate one or more events to it. The figure 6.23 shows that he selects the event icon:

A menu is displayed listing different event possibilities. The more obvious ones are the events directly deduced from the object 'Bewerbungsunterlage' and the activity name 'löschen' in the positive and negative form: 'Bewerbungsunterlagen gelöscht' and 'Bewerbungsunterlagen nicht gelöscht'.

As a second possibility, the adverbs which were already displayed in the wizard are listed. The statuses corresponding to other activities are ignored because the probability that they are connected to this process step is low.

The two other objects 'Bewerber' and 'Personalbedarf' involved in this process are also displayed. If they are selected, the statuses are displayed in a sub-menu, analog to the previous figure.

The menu allows multiple selections except from the two last menu points. The menu item 'Other objects' opens the wizard, without the steps 2 and 3. The last menu item description follows.

The process modeler selected the two first events, which are mutually exclusive. The figure 6.24 shows the generation of these two events joined by an 'xor' connector object and associated to the process step.

The last figure of the controlled modeling use case 6.25 shows the creation on the fly of an event:

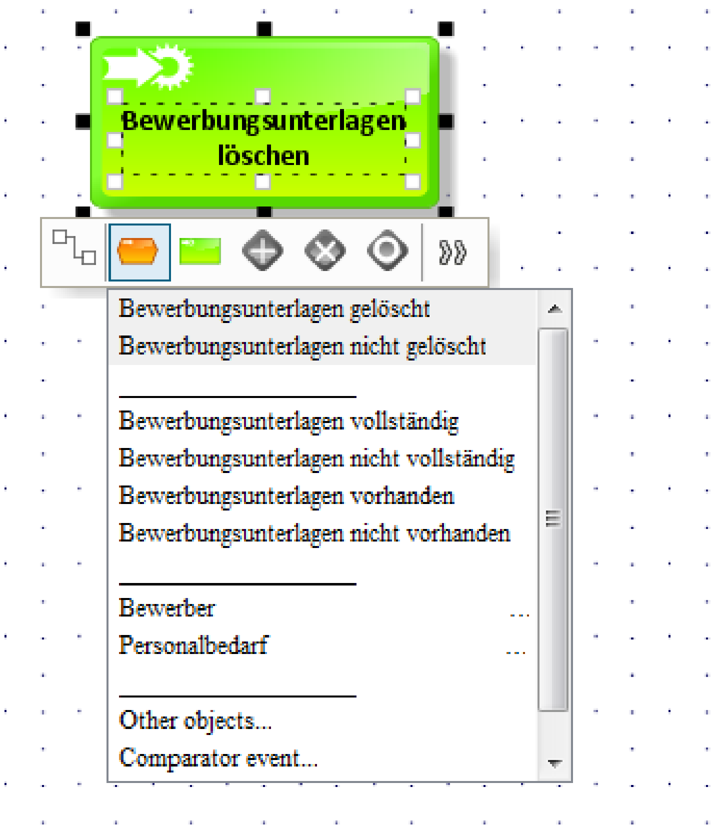


Figure 6.23: Controlled modeling: associate events to the created process step

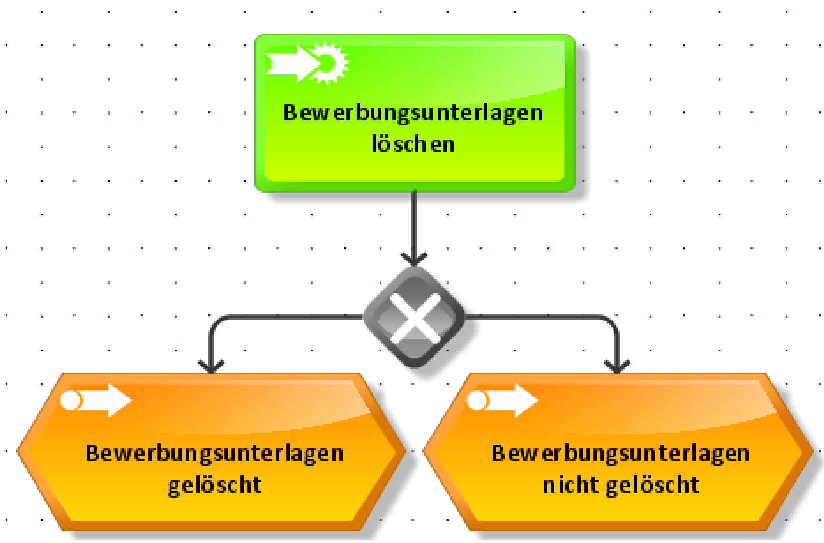


Figure 6.24: Controlled modeling: generated events associated to the process step

Analog to the process steps, objects involved in the process are listed with additional information in sub-menus (not displayed here).

The menu 'Other objects...' opens the wizard excepting the steps 2 and 3.

In this screen mock-up the process modeler selected the last menu item 'Comparator event...'. A window has opened. Comparator events are objects like 'kleiner 5 km', 'Bewerber > 50 Jahre'. The structure of such events is '[Object] comparator count [Unit]', where the object and the unit are optional.

When no object is specified, the check box 'No object specified' is selected.

The screen shows that the terms involved in this process can be directly selected. If another object is needed, the first wizard step will be displayed, allowing the selection of a term.

In the bottom of this screen mock-up, the process modeler can select the comparator, enter a count and select a unit. According to the data entered in this mock-up, the event '< 2 Wochen' will be created. Units must be registered in the knowledge base to allow their consistent usage (km, kilom., kilometer...). Sometimes the plural form of the abbreviation has to be used and the business content administrator must determine whether the long form or, if available, the abbreviated form shall be used.

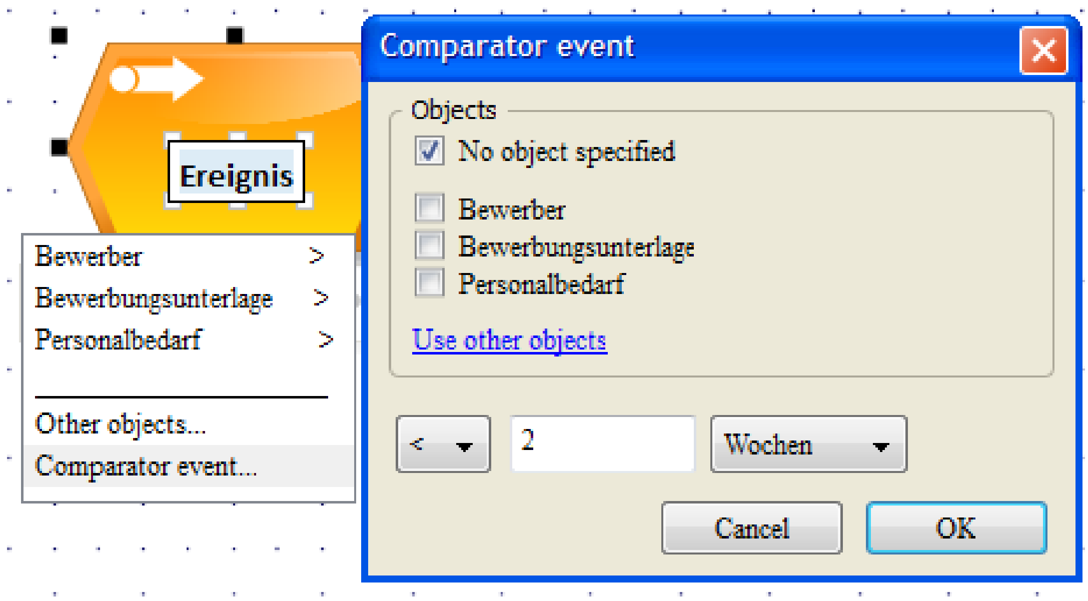


Figure 6.25: Controlled modeling: creation of a comparator event

More complex comparator events like 'Alter > 20 Jahre und < 50 Jahre' must be split in two events 'Alter > 20 Jahre' and 'Alter < 50 Jahre' and joined with an 'and' connector object. If the comparator events are standardized, they can be interpreted/transformed for implementation. If too many different cases have to be represented, the comparator events have to be transformed in a business rule which is able to represent complex events. The standardized comparator notations can further be used for business rules generation.

6.2.1.3 Quality Check Report

The last use case concerning the process modeling is a reporting use case. In the restricted and controlled modeling use cases, business modelers either correct on the fly their erroneous input or cannot enter erroneous object names at all. Assuming that the warnings fired during the restricted modeling are corrected, no further verifications are needed after the modeling. This assertion only counts for newly created business objects. However, companies often have legacy business repositories which were not checked against the modeling rules. The quality check report provides a complete business repository check. The business objects are checked against quality rules, terminology usage and spelling mistakes. The output consists in a report that summarizes all inappropriate object names. The erroneous objects can be corrected manually or automatically after a report validation if a correction proposition is available. Not all erroneous business objects have a correction. When spelling mistakes are available, a fuzzy match can be implemented to find an adequate term amongst the terms allowed in the context, but if an unregistered term is meant, the proposition will not be valid.

The figure 6.26 shows an extract of such a report:

> The report header displays the checked database names and the included knowledge base. The standard knowledge base is always used and the business content expert can include some domain knowledge bases (logistics, pharmaceutical industry, insurances...) or company specific knowledge bases (here 'user1'). These options are managed in the business content collaboration platform.
>
> The body of the report contains a table, summarizing all the warnings fired during the database check. Each line in the result table includes the business object identifier, which serves as a link into the business database.
>
> Comments about the sample results: no correction proposition is proposed for some of the marked objects.
>
> - The event object 'id98765' contains a spelling error.
> - Business objects including unspecific terms: in the example object 'id65432', the term 'Daten' is used with the activity 'anfordern'. The amount of data type that can be used with the activity 'anfordern' is too huge to find out which one is meant. Even if the objects deriving from 'Daten' are further filtered with help of the process map, it does not ensure that an unambiguous proposition can be generated.
> - Some object names like the object 'id43210' have such a complex structure that the quality checker can't interpret them. The modeler is invited to rename the object.

Quality check report

Business database: xyz_base
Knowledge bases: standard, logistik, user1

Object id	Object type	Object name	Message	Correction proposition
id09876	Event	Ablösungsvertrag ist vorhanden	Don't use 'sein' in events	Ablösungsvertrag vorhanden
--------	---------	-----------------------------	-----------------------------	-----------------------------
id98765	Event	Abmhanung senden	'Abmhanung' is unknown	
--------	---------	-----------------------------	-----------------------------	-----------------------------
id87654	Function	Email schicken	'schicken' is deprecated	Email senden
--------	---------	-----------------------------	-----------------------------	-----------------------------
id76543	Function	Mitarbeiter zu benarichtigen	'benarichtigen' ist deprecated Don't use 'zu' in functions	Mitarbeiter informieren
--------	---------	-----------------------------	-----------------------------	-----------------------------
id65432	Function	Daten anfordern	'Daten' is unspecific	
--------	---------	-----------------------------	-----------------------------	-----------------------------
id54321	Function	Leiter der Abteilung informieren	'Leiter der Abteilung' is deprecated	Abteilungsleiter informieren
--------	---------	-----------------------------	-----------------------------	-----------------------------
id43210	Event	Fahrzeug (Neuwagen) auf Lager	Do not use parenthetical expressions	

Figure 6.26: Quality report of business databases

6.2.2 Process Optimization Time

6.2.2.1 Process Consolidation

Process optimizing can be realized either in reducing the time to perform a process step or in eliminating redundant process steps. The latter is named process consolidation. If the business objects have unique and well-defined names, the redundant steps are easy to find, for instance with consolidation tools coming with the business process modeling environment. However, if the business repository was not elaborated with the semantically-enriched BPM applications, redundancies may remain: identical objects with different denominations. To support this step, process owners can execute a consistency check whose objective is to find out all redundancies. This check could be used in conjunction with the quality check report for quality purposes. The output of the consistency check can serve as the input for built-in consolidation tools.

The figure 6.27 shows the result of a consistency check as a report:

> Again, the report header displays the name of the concerned business database and the used knowledge bases.
>
> The report body consists in a table summarizing the inconsistencies. The inconsistent objects are grouped and separated with a dashed line.
>
> Comments about the sample results:
>
> - The first four lines are constituting one group of inconsistent events that have different structure. The preferred form is emphasized with the blue

selection mark.

- The second group contains an object using the forbidden 'durchführen' activity. The consistency check has to recognize that 'Verpackung durchführen' is equivalent to 'verpacken'.
- The last group shows an inconsistency in the function structure involving the activity 'prüfen'. The system must recognize that in both cases, the activity is 'prüfen' and the status is 'zuständig'.

Other types of inconsistencies which are not represented in the figure may concern synonym terms ('senden', 'schicken') or terms that have different grammatical derivations ('Bewerberunterlage', 'Bewerbungsunterlage').

Consistency check report

Business database: xyz_base
Knowledge bases: standard, logistik, user1

Object id	Object type	Object name
id12345	Event	Zutritt nicht möglich
id98765	Event	Zutritt unmöglich
id56789	Event	Kein Zutritt möglich
id23456	Event	Zutritt ist nicht möglich
------------------	------------------	--
id 34567	Function	Gefahrgutverpackung durchführen
id45678	Function	Gefahrgut verpacken
------------------	------------------	--
id67890	Function	Abteilungsleiterzuständigkeit prüfen
id78901	Function	Prüfen, ob Abteilungsleiter zuständig

Figure 6.27: Consistency check

6.2.2.2 Process Merging

A very similar use case concerns the merging of business processes. For example, when a company buys another one, business processes must be merged to achieve a successful and optimized integration by avoiding process redundancies. The 'Merge' functionality of business modeling environments supports the process integration tasks by searching for equivalent business objects, that means business objects which have the same denomination. However, different companies probably use different and either inconsistent terminologies that prevent from getting reliable results. Enhancing the 'Merge' functionality with semantic techniques supports the process owners in their optimization tasks by finding out similar objects with potentially different denominations.

The figure 6.28 shows the results of the enhanced 'Merge' functionality in form of a report. The functionality is started on a process model stored in a source database and checks for analog objects in the target database.

> The report header displays the name of the source model and the name of its containing business database issued from the parent organization. The target business database belongs to the subsidiary. The resources used during the execution of the 'Merge' functionality are listed.
>
> The body of the report contains two tables. The first one shows the matches between objects issued from the source model and functions issued from the target business database. Similar objects are organized in groups separated by a dashed line. The first group shows objects with a synonym located in a part of a compound noun: 'Kunde' is synonym of 'Abnehmer'. The three first objects matching 'Kundendaten' are exact matches. The fourth is an approximate match, because it cannot be determined which type of data is meant, the same is true for the last group. With the object 'Interesse prüfen' it is not ensured that the interest concerns the customer.
>
> The second table shows the processes concerned by the source objects. The first two objects have similar objects in the process model called 'epc-45', whereas the last object has similar objects located in another process.

To simplify the output design of the report mock-up, the merge functionality concerns only one process model. The same mechanisms can be applied to each model in the source database to obtain a complete image of the redundancies. The output of the merge functionality could serve as the input for built-in merge tools in the business process modeling environment.

The inverse check can be performed to find out which processes do not have equivalent processes in the subsidiary and reciprocally. Without semantic technologies, the output would consist of all objects which have a different denomination. With semantic technologies, only really different objects would be reported.

6.2.3 Process Implementation Time

Like described in the section 5.4, the objective of modeling a Service Oriented Architecture in the business environment is to better bridge the gap between business processes and IT processes. The following use cases are showing how to optimize the management of technical services in the business modeling environment and how to simplify the mapping between business process steps and services.

6.2.3.1 Service Annotation

In a top-down/bottom-up approach, the business modelers are modeling the business processes whereas the technical process modelers are managing the service repository in

Merge report

Business database source: parentorg_base, **model:** epc_77
Business database target: childorg_base
Knowledge bases: standard, logistik, user1

Source objects	Target objects
id234 Kundendaten anfordern	id098Abnehmerdaten anfordern
	id987 Abnehmerdaten anzufordern
	id876 Abnehmerdaten abrufen
	id765 Daten anfordern
-----------------------------	-----------------------------
id678 Kunde kontaktieren	id654 Abnehmer anrufen
-----------------------------	-----------------------------
id098 Kundeninteresse überprüfen	id432 Prüfen, ob Abnehmer interessiert ist
	id321 Interesse prüfen

Source objects	Target models
id234 Kundendaten anfordern	epc_45
id678 Kunde kontaktieren	
-----------------------------	-----------------------------
id098 Kundeninteresse überprüfen	epc_23

Figure 6.28: Process merging

order to associate services to process steps in a later stage. In a modeling environment, service annotation techniques consist in relating the business object representing the service with business objects which give detailed information about the service like capabilities, input data and output data in a dedicated model (see section 5.4.2).

The current use case shows the contribution of semantics in service annotations and management. Usually, the technical modeler imports a service (here an external WSDL-file) into the business repository with help of a wizard. The standard import wizard proposed by modeling environments is extended with additional components which exploit the natural language description coming with the service.

The figure 6.29 shows the first additional wizard step:

> The text field 'Original service description' displays the text included in the WSDL-file.
>
> The subsequent text field 'Business service description' displays an optional service description provided by the technical modeler, for instance when the service description is missing or incomplete. Both descriptions are treated in the

same manner: terms matching the knowledge base are highlighted in the text fields. The blue ones are considered as business descriptors and the red ones as non-functional capabilities.

The found descriptors are summarized in the group box 'Business descriptors' in form of check box items that are selected by default. The technical business modeler is then able to deselect descriptors which are not accurate. The terms 'Aktiendaten' and 'DAX' have been found in the knowledge base, whereas the terms 'Verkaufszeitpunkt', 'Preis' and 'Volumen' can be specified in more detail, that means that more specific occurrences are available in the knowledge base. The application found out that the service is about 'Aktien'. That enables selective term propositions. If the proposition does not fit, the technical business modeler can then deselect the item and propose manually descriptors in the next wizard step. The term 'Anfrage' was found in the knowledge base, but considered to be too unspecific to act as a service descriptor.

Services are usually composed of several operations. The operation interfaces are listed in the bottom of the screen mock-up. In this example, the service is issued from an external service provider, so that the operation names do not fit the technical terms which are associated to each business term. As a consequence of this, no descriptors are derived from the operation names. In a top-down approach, the interfaces of the service operations fit the technical names (see 6.2.3.3). By analyzing the operation names, the service can even be described with a higher granularity on operation level.

Once the possibilities of automatic service description are exploited, the technical business modeler can supplement manually the description like depicted in the figure 6.30:

The process map offers the possibility to pick out classified descriptors. By default, the current process map entry corresponds to the topic identified by analyzing the service description text, here 'Aktienhandel'.

The cloud displays the most common objects occurring in this topic.

Similar to the cloud layout, the list box located in the left-hand side of the mock-up filters all terms belonging to the category 'Aktienhandel'. In this example, the technical modeler expanded the node 'Aktien' as a type. It means that all available types of 'Aktien' are listed, but not business terms starting by the string 'Aktien'. The technical business modeler has selected the process map entry 'Aktienhandel' as a service descriptor. While this term is too general to address a particular process step, it favors a thematically classification which is useful for further discovery actions.

In the bottom part of the screen, the technical business modeler is able to select some non-functional capabilities issued from a standard catalog provided by the knowledge base. Because these non-functional capabilities were not found directly in the service description, the technical modeler has to know that the

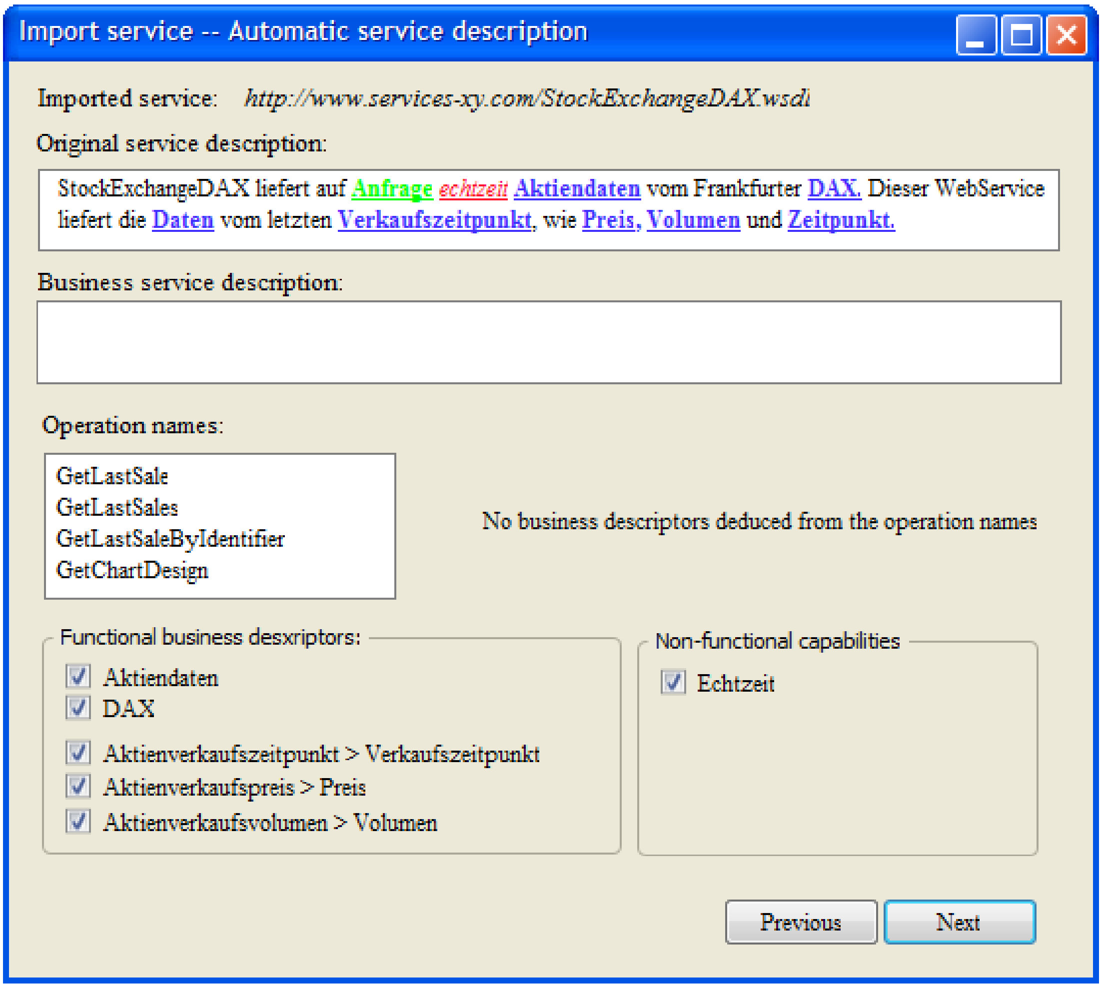


Figure 6.29: Automatic service description proposition

imported service really fulfills these capabilities. Such facts may be described, for instance, in marketing material.

If a needed capability is not available, it can be requested via the business content collaboration platform, analog to the business term requests in the use case 'Controlled modeling'.

The main difference between standard service annotation procedures and the described manual service description is that the service is annotated with a set of predefined terms. This standardization will increase the service findability.

At this wizard step, the technical process modeler has only selected a set of non-functional capabilities and a set of business terms which act as service descriptors. The wizard step depicted in figure 6.31 allows a deeper specification of the selected service descriptors:

The technical process modeler must specify for each descriptor whether it consists in a descriptor, an input data, an output data or a functional capability.

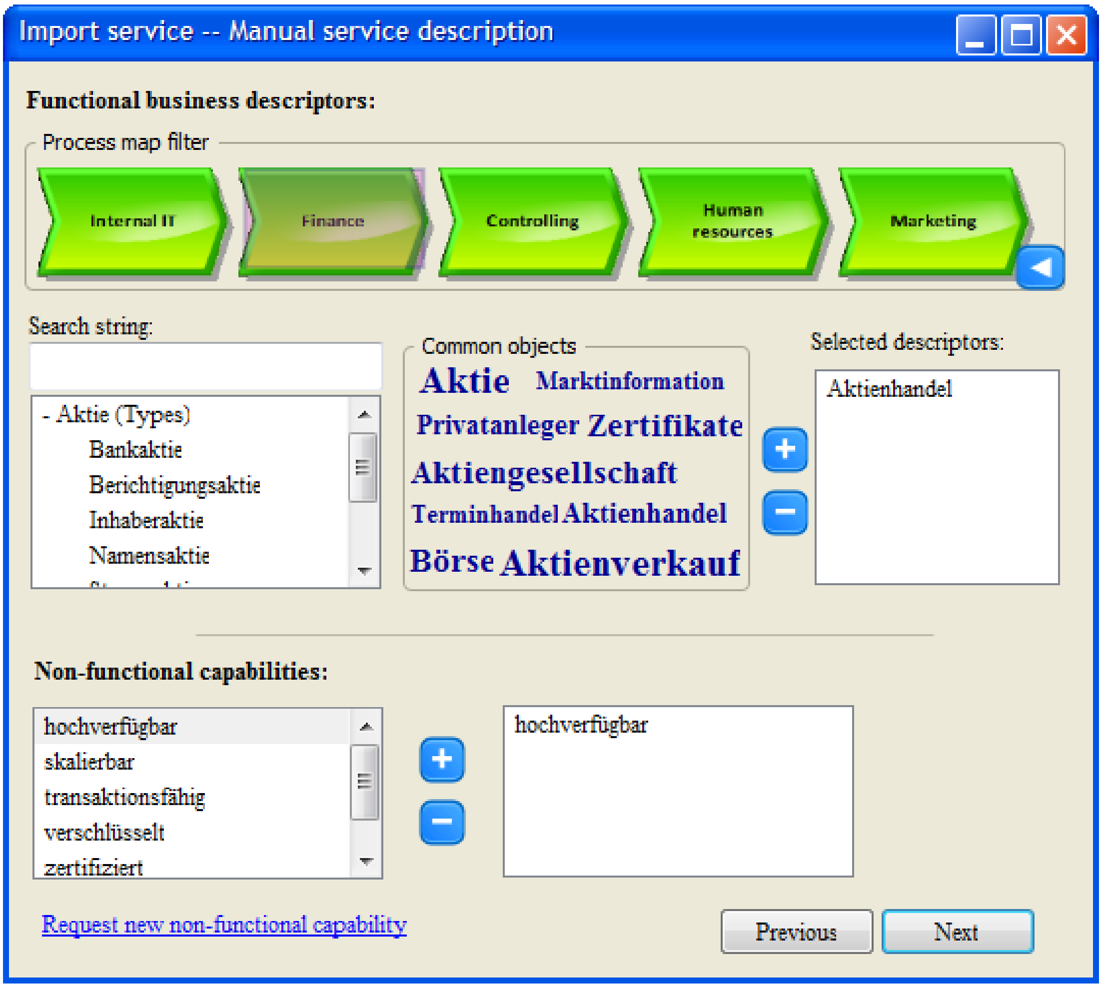


Figure 6.30: Manual service description

A descriptor allows a semantic classification of the service, like 'Aktienhandel', or gives more precise information about the service, like 'DAX'. However, it is not expected that these descriptors are modeled in the process model.

Input and output data are expected to be input and output of service operations and are sometimes modeled on a process step.

A functional capability modeled on a process step aims to describe semantically the requirement that a service must fulfill to execute the concerned process step. Thus, it is expected that the business descriptor is amended by an appropriate activity that denotes the action carried out on this object. In the example, one or several operations provided by this service are related to 'Aktiendaten abfragen'.

It appears that the descriptors can have different granularity. 'Aktiendaten' is a generic term which can group more specific data like 'Aktienverkaufspreis', 'Aktienverkaufsvolumen' or 'Aktienverkaufszeitpunkt'. In this case, it is correct to describe a service which provides several operations with more generic terms. It is then possible to associate more specific terms at operation level. This step is not depicted in the screen mock-up because the manual operation description procedure is analog to the service description one. The automatic part of the operation description is covered by the technical names topic.

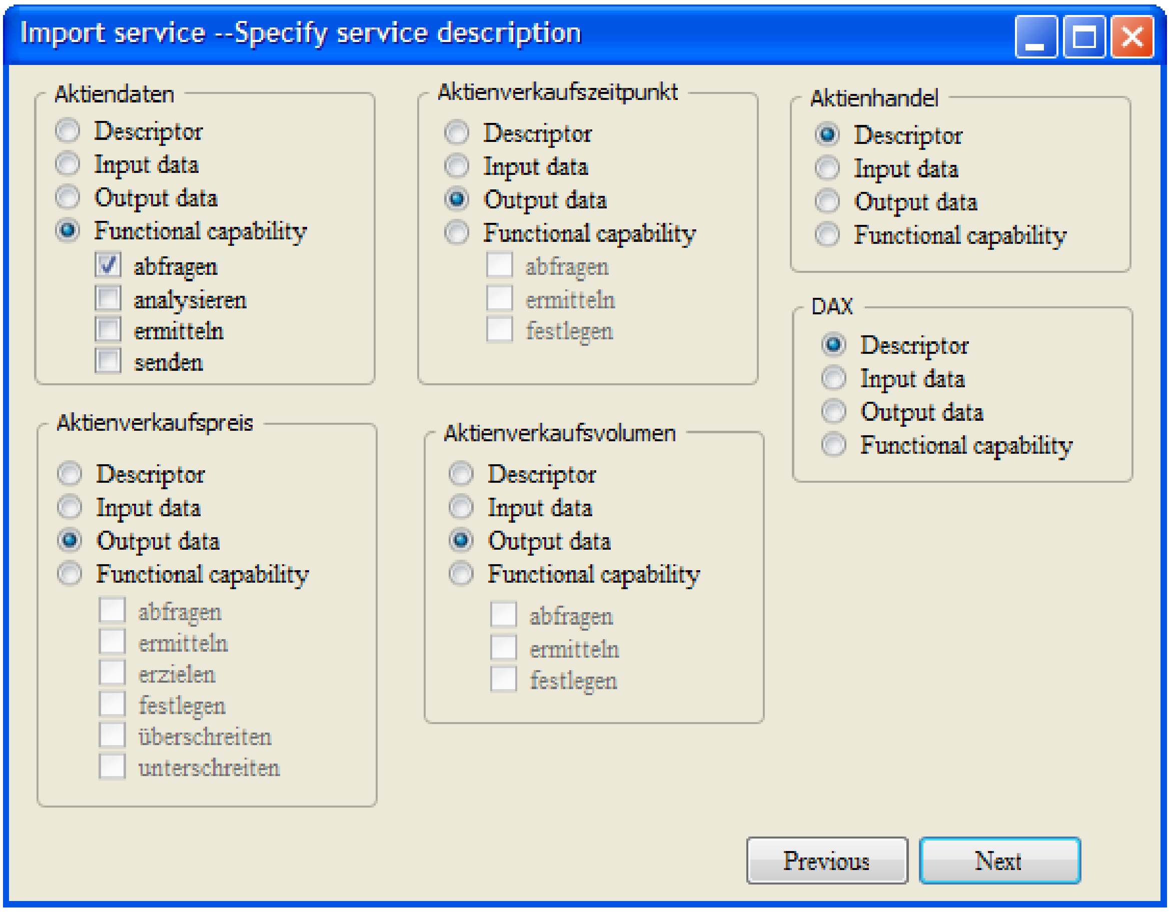


Figure 6.31: Further specification of service description

The figure 6.32 depicts the last step of the service import wizard:

> It recapitulates all service descriptors and allows the technical process modeler to switch back to previous steps if he wishes to change/complete some items.
>
> Once the service is fully annotated, he can finish the procedure and optionally generate a service annotation diagram in order to outline the service annotation visually and create the descriptors as business objects.

6.2.3.2 Service Discovery

Finding a service matching a process step is an interaction between requirements and capabilities. The service may be discovered with help of different criteria. At this stage, services are annotated with business terms and function names. Following search algorithm is performed during service discovery:

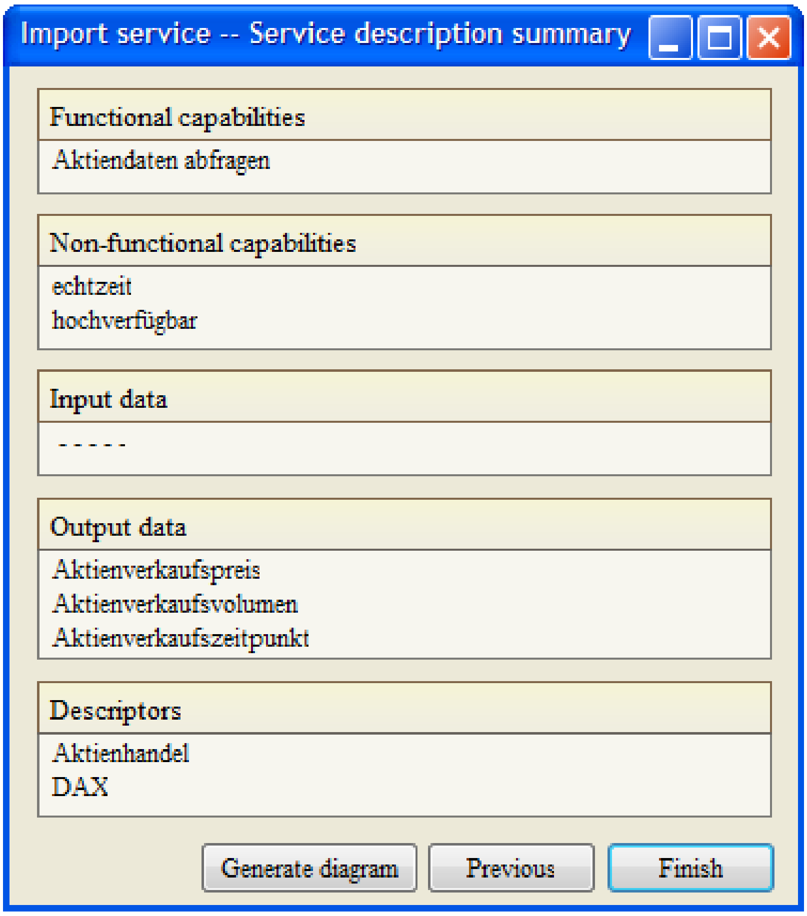


Figure 6.32: Summary of service description

1. Matching the name of the process step with service capabilities.
2. Matching the requirements related to the process step with service capabilities.
3. Matching input data related to the process step with service input data.
4. Matching output data related to the process step with service output data.
5. Matching the business terms used in the process step denomination with service input data.
6. Matching the business terms used in the process step denomination with service output data.
7. Matching the business terms used in the previous and subsequent process step context with service capabilities.
8. Matching the business terms used in the previous and subsequent process step context with service input data.
9. Matching the business terms used in the previous and subsequent process step context with service output data.

10. Matching service descriptors with an optional search string entered by the process modeler to perform a semantic service search.

The figure 6.33 shows a business process model concerning share trading:

> The point of interest is the process step 'Aktiendaten abfragen'. An output data 'Aktienverkaufspreis' is related to this process step. No requirements are modeled on this process.
>
> The process modeler can now start a service discovery within a service repository containing services annotated with previous techniques.

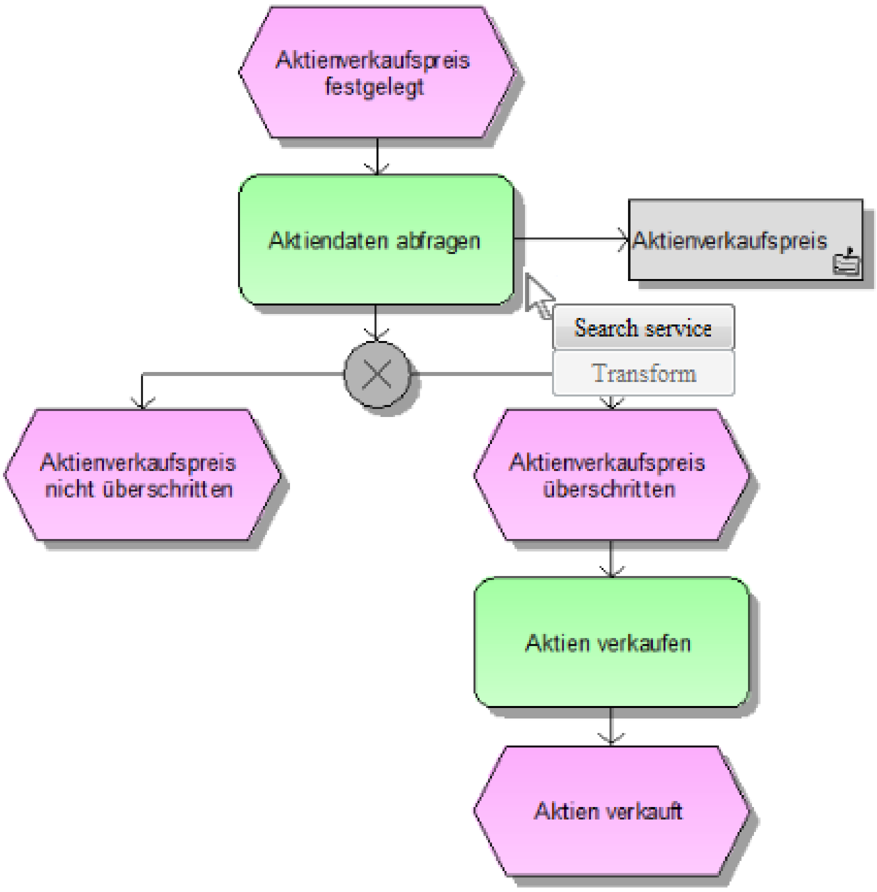


Figure 6.33: Search of a service for the process step 'Aktiendaten abfragen'

It is noticeable that the technical output term appears in the events modeled as results of the concerned process step. If no service is found with the standard algorithm (or on the contrary too many services found), it can be worth to examine the previous and subsequent process step context to refine the search. In the current example, the results would not have been affected by the context, because no additional information than the modeled information is available in it.

The figure 6.34 shows the result of the process discovery performed on the process step 'Aktiendaten abfragen':

> The search result proposes the 'StockExchangeDAX' as an appropriate service. The function name 'Aktiendaten abfragen' matched the service capability with the same denomination. If the function name is precise enough, the modeling of functional capabilities may be unnecessary, unless if needed for other purposes.
>
> The second matching criteria is the output term 'Aktienverkaufspreis'. All service annotations and the service description are summarized in this screen mock-up to allow the business modeler to view the other service properties and decide whether this service is really the one which fits best, especially for non-functional capabilities. For example, in share trading processes it may be very important to have a guarantee for high availability in order to monitor the share price evolution and take decisions at the right time.
>
> If several appropriate services are found, they are classified by matching criteria: the services with the highest number of matching criteria are considered as the most relevant ones.
>
> The link 'View technical properties' allows to explore the service operations, input and output parameters on a technical level.

In addition, here the granularity is a point of interest. It can happen that the terms used in the process step are more unspecific than terms used in the service description. In this case, more specific services must be displayed and the modeler must choose which service corresponds better. He may even correct the process step denomination with a more precise one. In contrary, the process step can be more fine-grained. Instead of 'Aktiendaten abfragen', the process modeler may have modeled 'Aktienverkaufspreis abfragen'. It is why the service search algorithm includes the steps 5 and 6, which match the process step denomination with input and output data. If the knowledge base or a technical term model reflects that 'Aktiendaten' and 'Aktienverkaufspreis' are related, the search procedure would have come to the same result without the output data modeling and without context examination.

6.2.3.3 Process Transformation into an IT Solution

The section 5.4.5 explained the problems encountered while transforming a business process into an IT process and during a round-trip scenario. The solution is based on mapping each business term with a technical term, for instance an English translation because English is one of the standard programming languages, in particular for companies who are outsourcing their software development. When the business process model is transformed in an executable process (for instance into BPEL or XPDL), the business terms occurring in functions and events are mapped into their corresponding technical terms and joined together according to specific joining rules. The resulting function and event names are then consistent and can serve as service interfaces for further implementations in development environments.

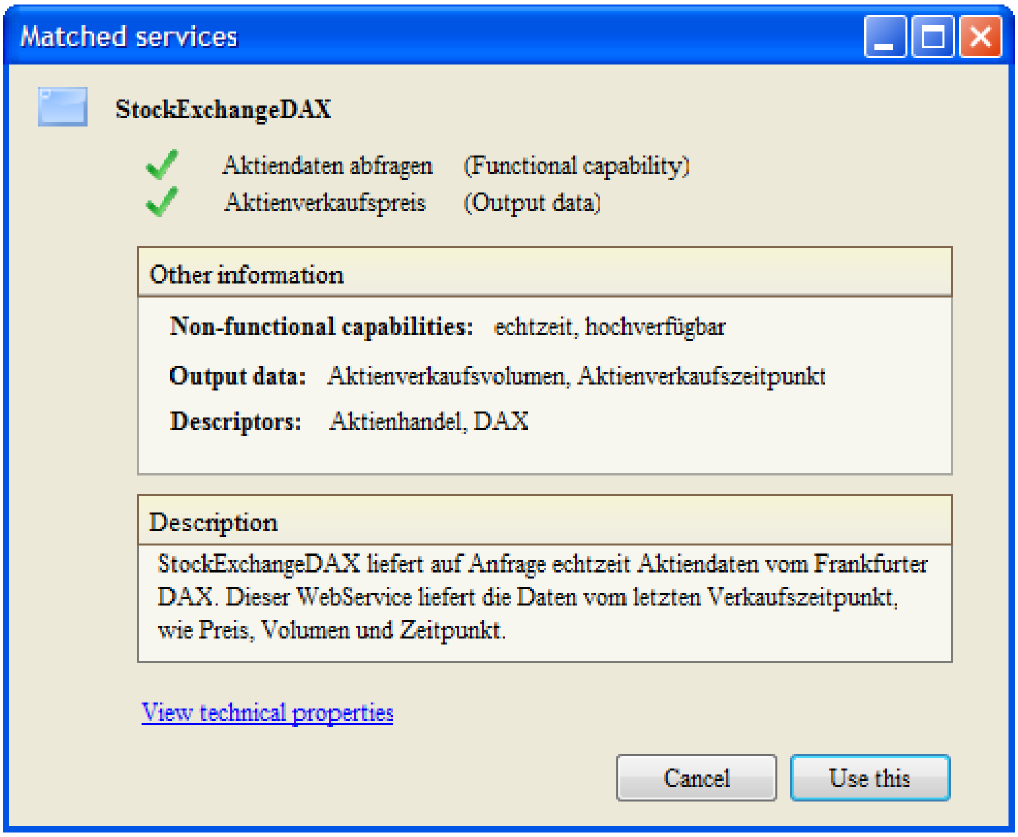


Figure 6.34: Matched service

In a round-trip scenario, the IT process is supplemented with technical information and reimported in the business modeling environment. Sometimes additional process steps are added directly to the development environment. When reimporting the IT process into the business modeling environment, the added process steps are only available in the IT model, so that the alignment between business processes and IT processes is interrupted. If the technical interface name can be correlated to a business function or event name, the additional step can be added not only in the technical model like BPEL, but even on higher levels like EPC or BPMN.

The figure 6.35 shows the process model transformation from an EPC model to a BPEL model. Some technical objects like scope begin, scope end or assignments were deleted in this model to simplify the example and emphasize the important objects. It is worth to note that the technical terms generated in this model do not fit standard naming conventions for software implementation where functions names often start with 'get' or 'set' (like the operations of the 'StockExchangeDAX' service). However, it is to consider that these names are acting as interface names for services and that the underlying implementation still can respect standard or corporate conventions.

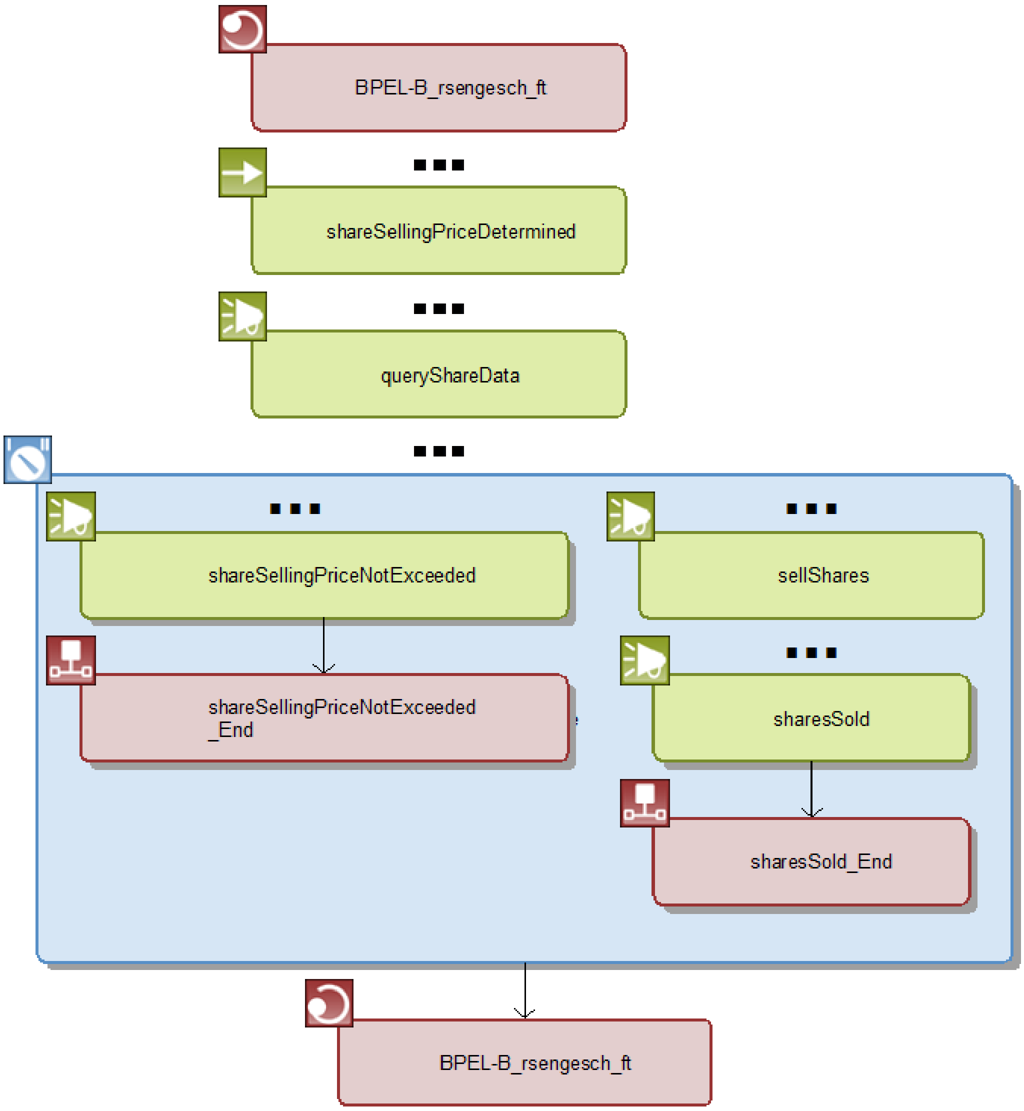


Figure 6.35: Transformation with introduction of technical terms

6.2.4 Process Execution Time

Process execution can mean two different things: the automatic execution of processes supported by software or hardware systems and the execution of manual tasks through process participants. The latter is point of interest of the two following use cases.

6.2.4.1 Retrieval of Published Processes

After the processes are modeled, a company often publishes them in the intranet to allow the consulting of processes by process participants. If process participants are not aware of a particular process, they may want to find out which process steps they have to perform to execute a task like reporting travel expenses. Today, collaboration platforms or tools for publishing processes are allowing the search against tags entered manually, complete object denominations or a classical string search against object names. No integrated retrieval mechanism is available that allows to search with synonym terms or with terms deriving from the search string.

The figure 6.36 shows a business process model published in a business process collaboration platform and an integrated search mechanism that uses semantic techniques:

> The collaboration platform consists of a portlet where models are displayed and of a portlet allowing to search for models. Other portlets specific to collaboration platforms are blinded out.
>
> The process participant has searched for processes involving the term 'Bewerbersunterlagen'. As an exact search result, the process involving 'Bewerbungsunterlage' is displayed. Two neighboring processes concerning the terms 'Bewerber' and 'Bewerbung' are also proposed.

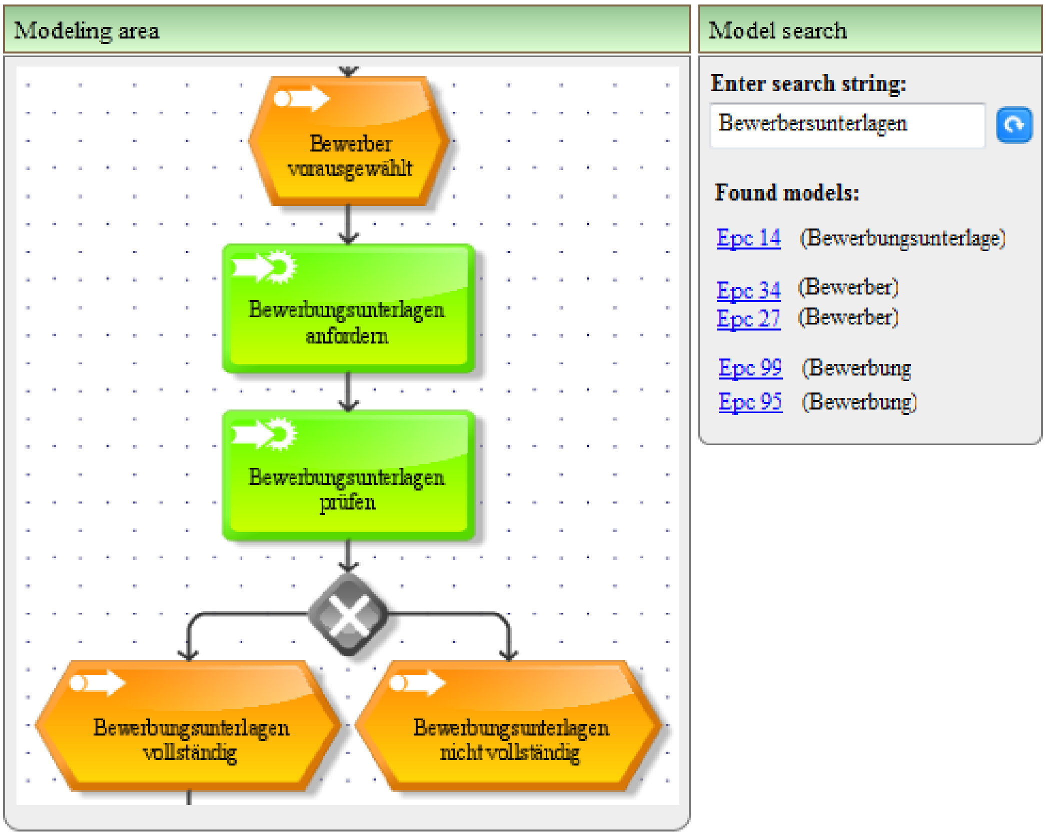


Figure 6.36: Process search in a business process collaboration platform

6.2.4.2 Tagging of Business Processes

To favor process model findability in a collaboration platform, the process models should be tagged before uploading to the intranet. Tags either are entered manually or consist of the automatic extraction of occurring business object denominations. Entering tags manually costs expensive human effort while generating tags from entire business objects denominations is not precise enough: the process participants may never enter a whole process step name into the search field. With semantic techniques, business terms are extracted from business objects, transformed into their citation form and, if necessary, mapped to adopted synonyms.

The figure 6.37 shows a published process model in which tags have been generated automatically with the help of semantic techniques:

> The generated tags are displayed in the 'Model description' portlet. The terms involved in the business process are extracted and displayed as tags after mapping to their citation form. In the example, the plural form is reduced to the singular form. If term variants were used in the process model, the standardized form would be displayed as a tag name ('Bewerberunterlage' vs. 'Bewerbungsunterlage').
>
> The user keeps the possibility to enter tags manually. These tags will also be reduced to their citation form and transformed in the adopted standard form to favor a better process model findability.

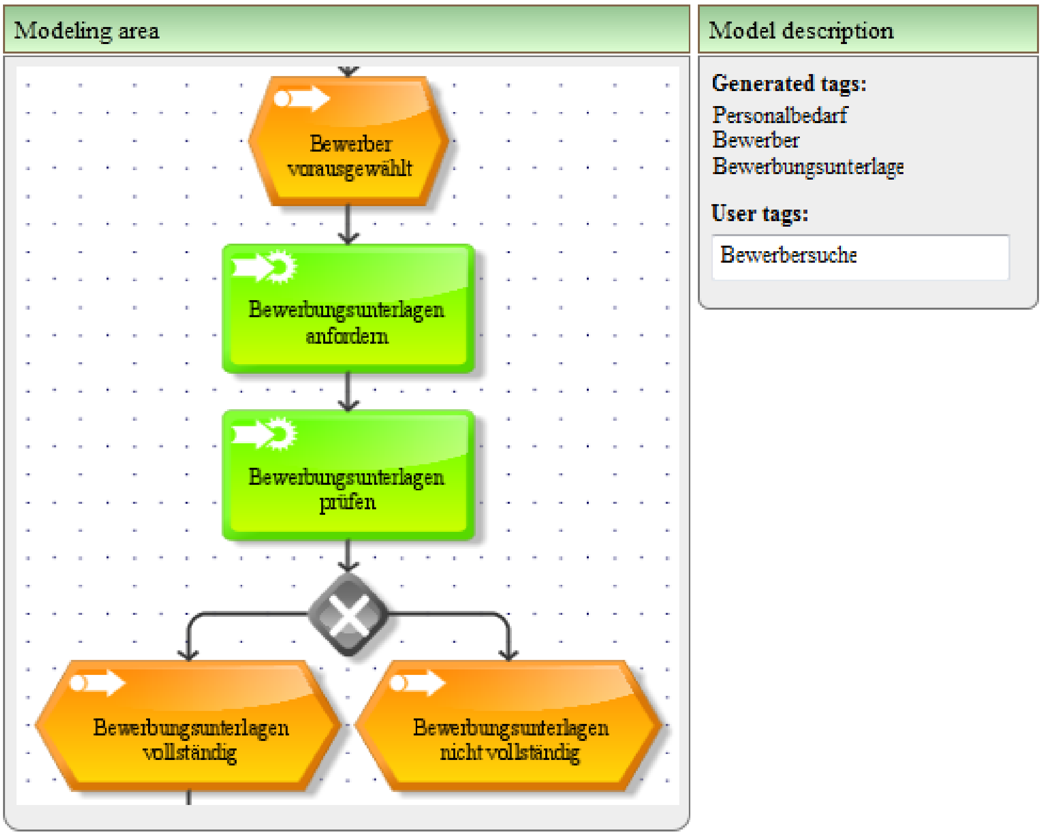


Figure 6.37: Tagging of processes

6.2.5 Process Design Time

Although process design is the first phase of the process life cycle, the use cases concerning the design time are described only now because the other phases offer a clearer introduction to the benefits of semantic techniques in BPM. On the other hand, process management is a life cycle whose aim is continuous process improvement so that the design phase begins again after process optimization or when new processes have to be modeled.

6.2.5.1 Impact Analysis

When optimizing a process, the process modeler has to evaluate the impact of modifying a process step on other processes. If the business repository has been elaborated with the controlled modeling application, the impact analysis is quite simple: it consists in retrieving business objects with the same denomination. If not, the use of semantic techniques allows seeking for all object variants like in the 'Process Merging' or 'Process Consolidation' use cases.

The figure 6.38 shows a search window enhanced with semantic techniques, which is integrated in a business process modeling environment:

The object name to search for can be entered manually.

Alternatively, the search functionality can be started by selecting a specific business object.

In the present figure, the option 'Exact match' is selected. Indeed the results have variant denominations, but they are semantically equivalent.

If the business modeler wishes a standard string match, he must use the standard search functionality provided by the modeling environment.

The figure 6.39 shows the same search functionality as in the previous figure, but having the option 'Exact match' deselected:

Additional objects with an approximate semantics are displayed.

In the previous figure, a negative event is searched and only negative events are displayed. In the present figure, a positive event is searched and negative events are displayed as well. The event containing the unspecific term 'Unterlagen' is also suggested.

6.2.5.2 Document Classifying

Like seen in the chapter 5.2, the process design is frequently based on written documents like process descriptions or interview sheets. The documents may be uploaded in content management systems or stored on devices, without an appropriate classification allowing a quick finding of accurate documentation. The

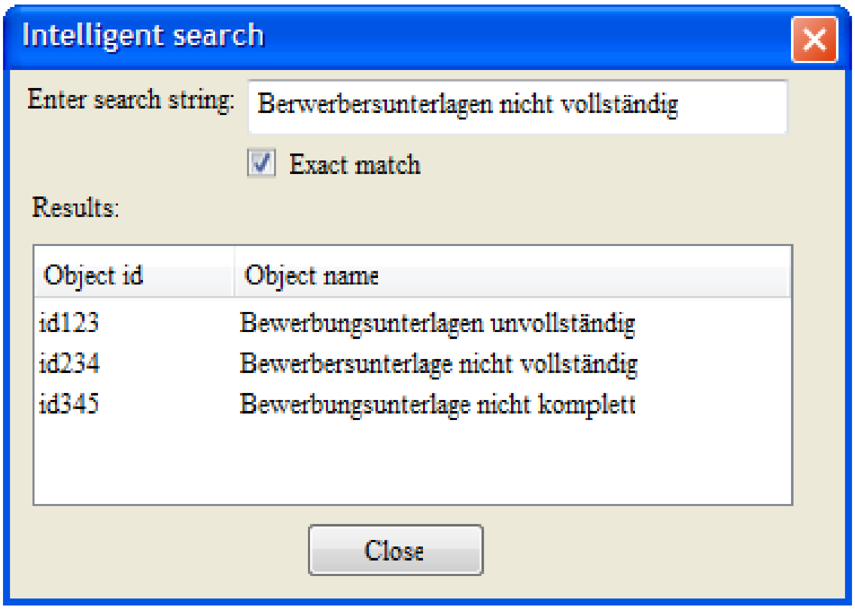


Figure 6.38: Searching for an exact denomination

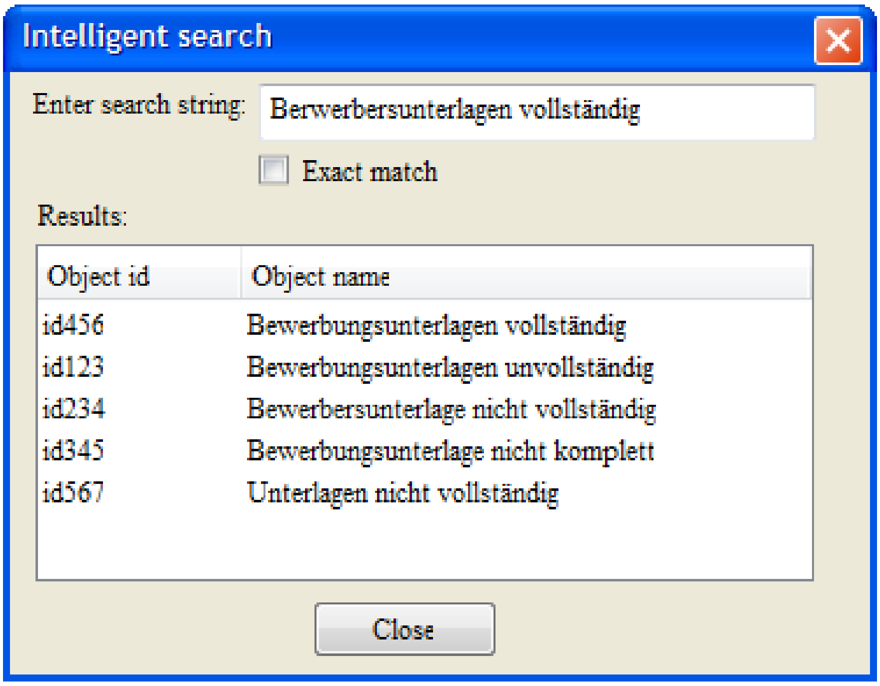


Figure 6.39: Searching for an approximate denomination

corporate business collaboration platform can be coupled with existing content management systems. The 'Document Classifying' use case aims to categorize process documentation into process map elements or to map them directly to a given process step. Documentation is classified according to the occurring terminology. With help of statistical functions basing on the business terms usage and their occurrences, the documents are classified in more or less detail in the map. The classification values are a mix of values issued from the standard knowledge base and from process models stored in the business repository. When process modelers are searching for appropriate process documentation, they can navigate to the process map elements and find the available documents.

The figure 6.40 shows the corporate business content collaboration platform with the business content expert administration rights:

> On the left side, the 'Tasks' portlet shows the open tasks to be processed by the business content expert and the portlet 'Content management' provides the management functionalities accessible to him.
>
> The central portlet displays the results of a documentation classification procedure. Documents are associated to a process map element. The most frequently used and most relevant business terms are displayed below the document name.

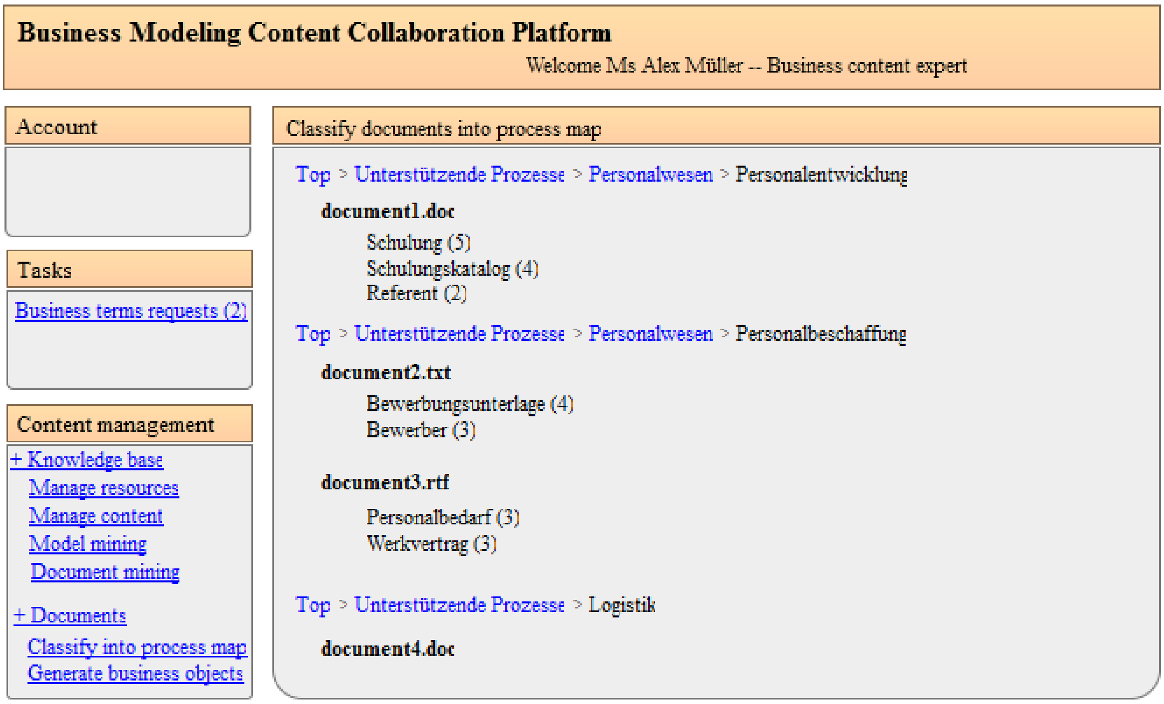


Figure 6.40: Classifying documents in process maps

The analog use case which consists in mapping documents directly to a business model or to a process step is not depicted here.

6.2.5.3 Document Mining

Once the document is classified, the process modeler selects appropriate documents and starts the modeling. In the majority of cases, the documents are not checked against terminology usage, so that the process modeler must cope with different vocabulary. He must transfer written sentences to business objects. The use case 'Document mining' supports the process modeler in this task.

The figure 6.41 shows the setting up of the mining task:

> The business content expert or the process modeler selects the documents describing a process and binds the needed knowledge bases.
>
> The business content expert decides which content he wants to share with the community: business terms, activities and context.

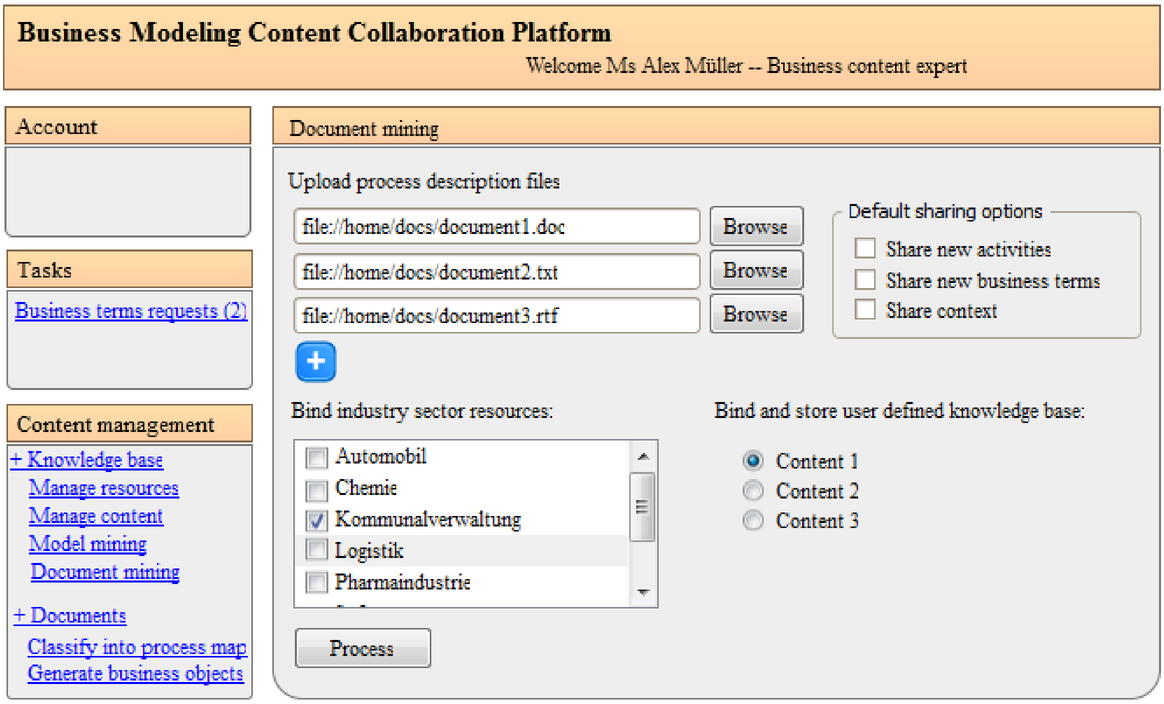


Figure 6.41: Setting up the mining of business objects in documents

The figure 6.42 shows the results of the business object mining procedure applied to the document 'document1.doc':

> The table summarizes all objects which can be generated in a business model.
>
> - The first table column contains check boxes that enable or disable the object generation for each table line.
> - The second column lists the object types.
> - The third column contains the generated object names.
> - The fourth column contains the sentences from which the business object names are mined.
>
> Comments about the results:

- In the second table line, the word ’updaten’ is replaced by ’aktualisieren’.
- The third and the fourth line list two events generated from the same sentence.
- In the last table line, ’TrainSys’ is interpreted as an IT system.

In the bottom of the portlet, the sentences from which no objects are generated are listed. The process modeler must create the objects manually and maybe update the document to reduce the complexity of the sentences to make them more readable for humans and for machines.

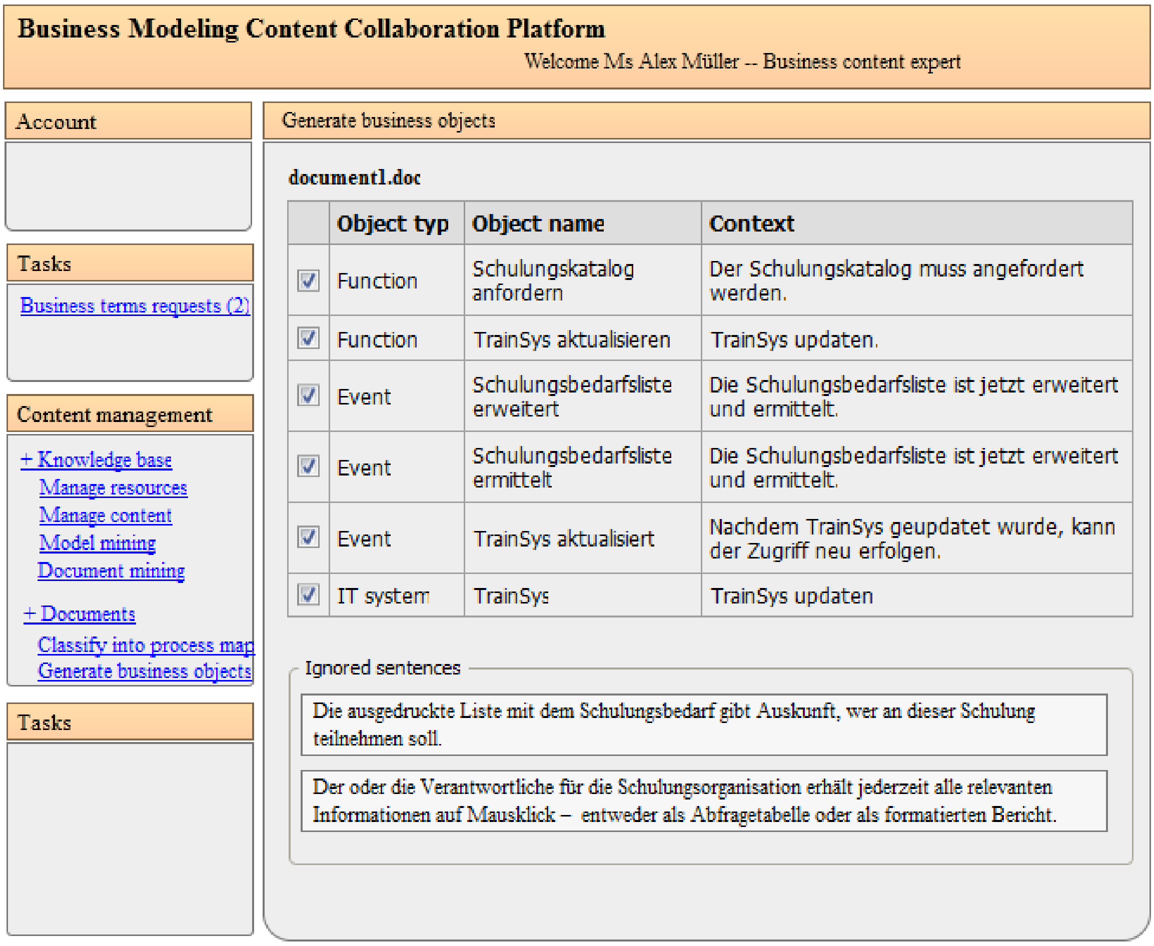


Figure 6.42: Mining business objects in documents

6.2.5.4 Document Alignment

A complementary use case supports the change management process. Documents are compared to process models in order to detect process changes. The objective is to reach a perfect alignment between process models and process documentation. Two types of business objects can be detected:

- Business objects generated from the document which are not part of the process model
- Business objects which are part of the process model and are not occurring in the document.

The first case shows that the document was updated and not the model. The second case shows that either the process model was updated and not the document or that some parts of the document were deleted without updating the process model. The output is analog to the output of the use case 'Document Mining'.

6.2.6 Business Process Content Governance

BPM governance denotes the process of process management. All described use cases rely on preset and extensible business content. The quality of the semantically enriched BPM applications directly correlates with the quality of the knowledge bases, which must be exhaustive, reliable, accurate, sustainable and restrictive, to achieve a consistent business repository. At the governance level, a business content expert is responsible for maintaining the business content and for assigning the needed content resources to the process modelers who are performing the BPM tasks. The next use cases deal with the management and maintenance of the business content.

6.2.6.1 Resources Configuration

The knowledge bases to use are dependent on the domain or branch of the company. To avoid high management effort, a standard knowledge base and a set of standard industrial sector knowledge bases are provided by the content collaboration platform. As a first step before using SE-BPM applications, the business content expert configures the involved resources.

The figure 6.43 shows a central portlet where different SE-BPM projects can be administrated:

> A company may be divided in several branches so that for each branch different resources are needed. It is why the resource configuration is organized with projects, which consist of a set of knowledge bases. The menu in the upper right corner shows that a project can be created, opened, saved and deleted. Deleting a project does not delete the knowledge bases, but only the references to them. The project name is displayed in the header.
>
> Underneath, a selected and disabled check box shows that the standard knowledge base, here called 'Build-in library' is always included and cannot be removed.
>
> Preset domain-specific knowledge bases are selected in the industry sector resources group box. The right list box displays the resources selected by the business content expert.

In the group box located at the bottom, he can include company specific resources whose management is explained in further use cases.

The industrial sector resources content is overwriting the standard library and the corporate specific content is overwriting the industrial sector ones.

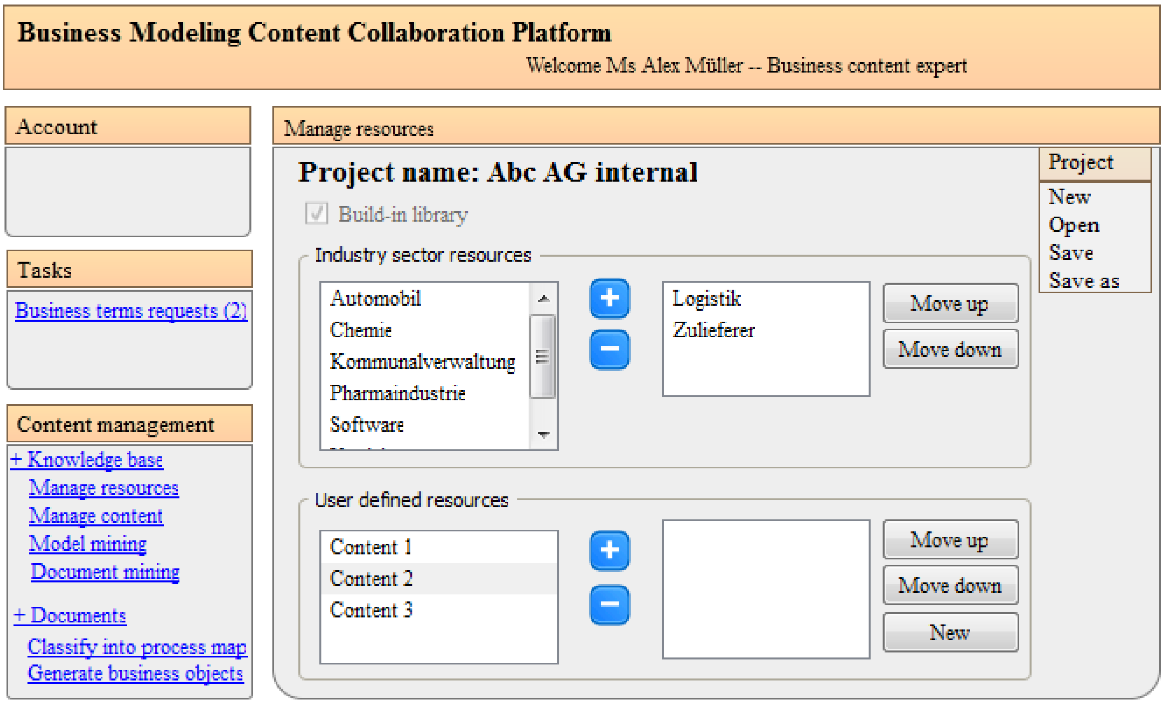


Figure 6.43: Selecting the resources for SE-BPM applications

6.2.6.2 Manual Management of Business Knowledge Bases Content

If only preset business knowledge bases are employed, all SE-BPM applications can be applied with exception of the controlled modeling. Only restricted modeling can be adopted to support the modeling task and in the other use cases, one must count with a quality loss when encountering unregistered terminologies. The three following use cases are dealing with the administration of knowledge bases.

Theoretically, business knowledge bases could be administrated manually. This is not the favored way because language is composed of an infinity of possible terms (compound nouns or noun phrases) and thus manual administration would be very counterproductive: business content expert would manage terms that would probably never be used in business process models. While a complete manual administration is not reasonable, some manual adjustments may be useful. Not the complete administration component mock-ups are depicted here, but two meaningful examples.

The figure 6.44 shows the property page of the activity 'senden':

The 'Properties' panel displays the concept name 'TO_SEND' and its technical term 'send' used for the transformation of a business process into an IT process. The next chapter will deal with concepts in more detail. The status corresponding to 'senden' is 'gesendet'. When activating the according link, the property page of the status 'gesendet' will be displayed. 'senden' is the preferred denomination and is not abbreviated.

The 'Synonyms' panel displays a list of synonyms, which are all deprecated. These common activities are part of the standard knowledge base. They can be overwritten (with another preferred form, additional synonym) and saved in a company specific knowledge base. The standard libraries never will be altered.

On the right side, the panel 'Business expression structure' is displaying restrictions applying to the activity 'senden'. The structure of process steps involving the activity 'senden' requires an object, an optional agent (the receiver of the object) and an optional activity modifier. In this example, the objects to send are restricted to 'Brief' and 'Email'. The receivers are not restricted. The activity modifier can be 'erneut'.

The property pages can differ a little bit according to the object type: activity, statuses, objects, units... For instance, adverbs acting as statuses like possible ('möglich') have a link to their inverse status like impossible ('unmöglich').

The figure 6.45 shows the object hierarchy of the term 'Vertrag':

The displayed hierarchy corresponds to the context 'Personal Procurement'. If the root context were selected, the object amount would have been greater.

Each term is placed in a table whose header consists in a link leading to the detailed property page like the previous figure. The tables are summarizing the object relations. In the table 'Vertrag', the first column lists the object modifiers: the object 'Vertrag' can be modified by the modifiers 'neu' or 'alt'. The object modifiers can be used in business objects denominations if the activity structure allows it: in the previous figure, the activity 'senden' has no optional object modifier specified, so that the modifiers 'alt' or 'neu' cannot be used with this activity. Not all of the more specialized objects listed on the right side of the parent object have the modifier column displayed: they inherit the parent modifiers. The object 'Arbeitsvertrag' comes with two additional modifiers, 'befristet' and 'unbefristet'.

The second column of the table 'Vertrag' lists the activities which can be associated with the object 'Vertrag' and with its child objects. All child objects have an additional activity relation, 'anbieten'. Some of them have the activity 'korrigieren' disabled (by clicking the arrow button).

The last column displays the relations between the object and the statuses. In the table 'Vertrag', only the adverb 'vorhanden' is displayed. Statuses derived from the activities are by default activated and not shown here. The child

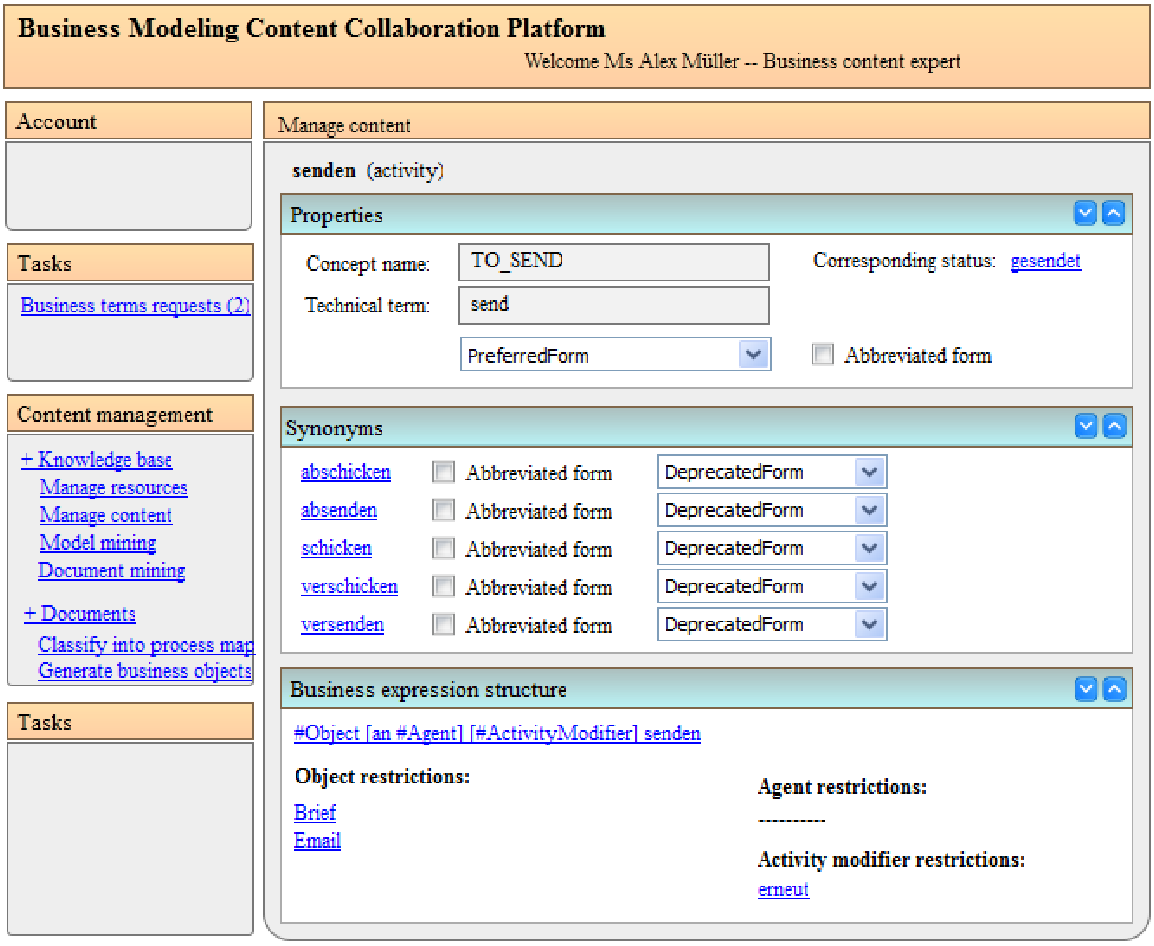


Figure 6.44: Properties of the activity 'senden'

objects 'Abwicklungsvertrag' and 'Aufhebungsvertrag' are associated with two verb statuses, 'akzeptiert' and 'abgelehnt'. These statuses are displayed here because the activities 'ablehnen' and 'akzeptieren' are not coupled with these objects or with the parent object.

6.2.6.3 Business Database Mining

As soon as the business content expert decides that the process modeler must apply controlled modeling instead of restricted modeling, all terms needed by the process modeler must be registered in knowledge bases. Like described in the previous use case, administrating business content manually would be too expensive. The SE-BPM applications must provide facilities to support the business content expert in this task by reducing the administration efforts to the minimum.

With restricted modeling, the object denominations are only checked against spelling mistakes, structure rules and standard terminology and relations. New terms will not be taken into account. When switching from restricted modeling to controlled

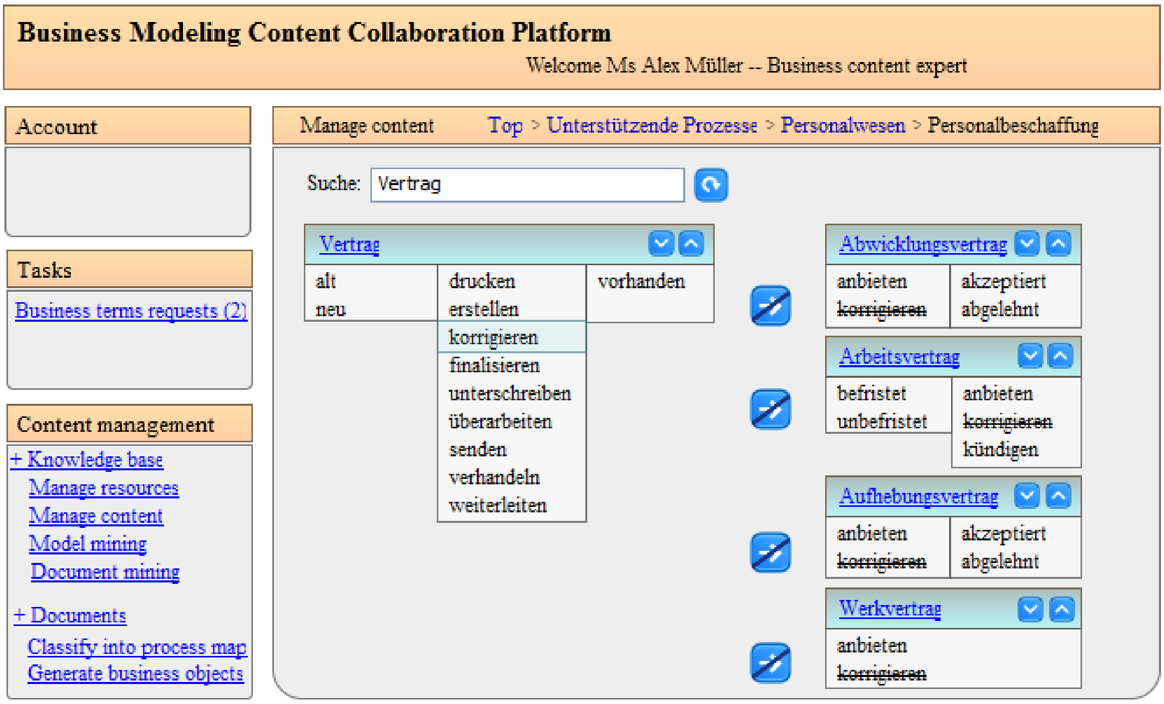


Figure 6.45: Hierarchy of the term 'Vertrag' in personal procurement context

modeling, legacy databases containing unregistered business terms are available. Objective of the business database mining is to register the used terms and maybe correct them or propose an alternative. Words marked as a spelling mistake are extracted as well because they can be an inappropriate abbreviation, a technical term or a foreign language term. Once registered, the term will be known in the future.

The figure 6.46 shows the setup of the business database mining procedure:

Term mining can run on whole business repositories or on single models.

Again, the industrial sector and company specific knowledge bases can be used. The standard one is always available.

The results are stored in the selected user-defined knowledge base.

The business content expert can decide to share some of the results with the community.

The figure 6.47 shows an extract of the mining result concerning the status 'verheiratet', which was not registered so far.

In the result header, the base form of 'verheiratet', which is the activity corresponding to the status, is displayed. The first group box displays the contexts that are the business object denominations in which the found term appears. The check box located right to the context is selected by default when contexts are compliant to the readiness rules. As a consequence, the context

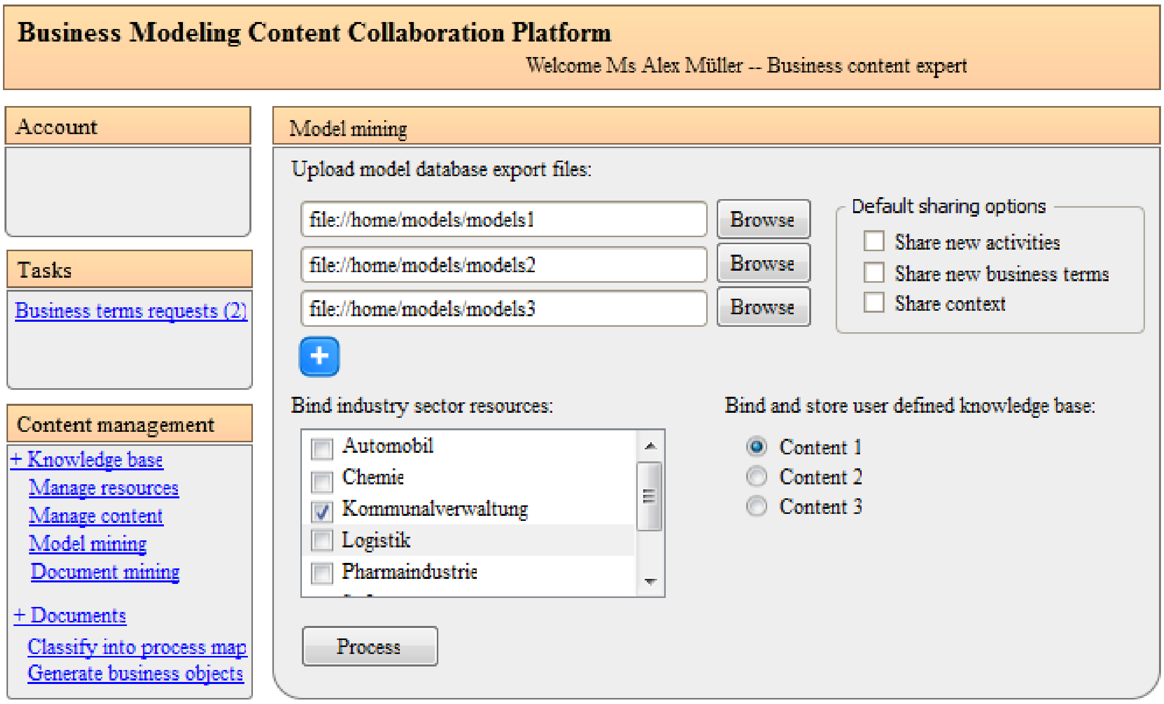


Figure 6.46: Business database mining setup

can be saved as the default business object denomination involving the terms 'Antragsteller' and 'verheiratet'. The business content expert has the possibility to correct the context, which leads to a re-check of the denomination to ensure that the stored contexts are respecting the quality assurance rules.

For each context involving a different business term, a potential synonym list is displayed on the right side. In the example, all activities related to the term 'Antragsteller' are displayed because one of the activities could be semantically neighbored to the extracted one. Three cases can occur:

- No synonym is selected for a given context: a new term entry and context entries will be created in the knowledge base.
- One synonym is selected: the extracted activity/status will be declared as a deprecated synonym of the selected check box. To change the extracted term to the preferred form, the business content expert must switch to the detailed property page. The probability that an extracted synonym term is the deprecated form is higher because it is less frequent.
- Several synonyms are selected: the extracted activity/status will be declared as a deprecated synonym of all selected check boxes, but can never be changed to a preferred form. Such terms do not have a strong and unambiguous semantics.

The status form of the activity is displayed under the context group box. The status form is not editable because it was extracted as is. If an activity is extracted (for example 'verheiraten'), the system searches whether the corresponding status form is available in the corpus and proposes the result. If not, the status form is entered manually and automatically verified: the base form of the entered status must correspond to the activity.

The technical term is necessary for the model transformation use case in a SOA. The 'Technical term' field corresponds to the technical term of the activity and the 'Technical passive term' field corresponds to the technical term of the status. These fields must be filled in manually if no synonym is selected. Otherwise, the technical terms of the synonyms are used. The concept name (see next chapter) derives from the technical name.

The check box 'Add relation Antragsteller - verheiratet' is selected by default because it corresponds to the context. However, the business content expert must confirm the relation between 'Antragsteller' and 'verheiraten', because no such context was found. If the check box 'Ignore verheiraten' is selected, the extracted term is stored in a stop word list. That means that the term is erroneous and has not to be extracted again in the future.

The business content expert can choose to share the mined term and contexts with the community to outsource the term management efforts. The check boxes are selected or deselected by default according to the global sharing options.

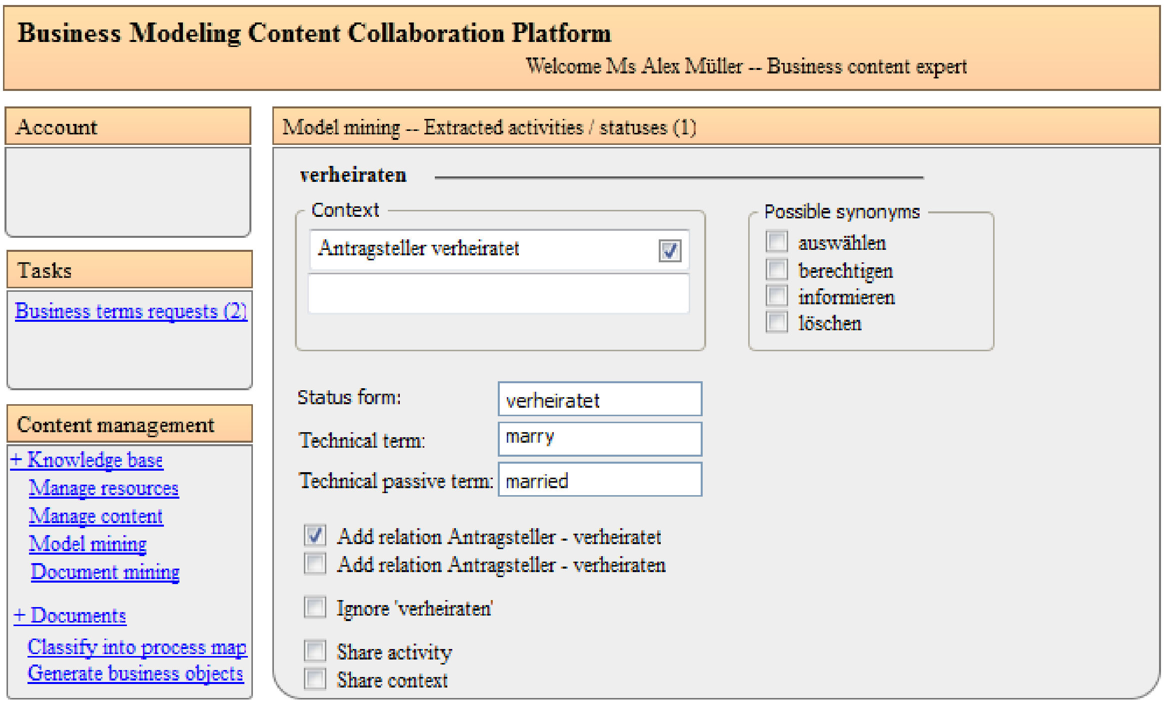


Figure 6.47: Model mining results: new activity extracted

The figure 6.48 shows an additional extract of the mining result, concerning the business object 'Schulungsbedarfsliste':

The term 'Schulungsbedarfsliste' is not part of the knowledge base, but the compound 'Bedarfsliste' and the term 'Schulung' are. All suggestions are based on the sum of these compounds. No potential synonyms were found.

The business content expert can edit the field 'Preferred form', which allows him to enter a synonym of the extracted term. The extracted term is then stored as a deprecated variant. The manual synonym input will be further analyzed to find out whether he entered a spelling mistake or whether a variant of the proposed synonym is available. These step by step verifications ensure a consistent knowledge base.

The technical term results in a concatenation of the technical terms 'requirementList' and 'Training'. The plural form of the technical term is proposed as a concatenation of the singular term 'training' and the plural head 'requirementsList'. The business content expert can edit the technical term propositions if they are not correct.

The system proposes a default classification in the topic map that can be changed by the business content expert. The classification proposal is based on the classification of the compound parts and the term usage in the business repository.

The check box 'Inherit all relations from "Bedarfsliste"' allows to associate the same activities, statuses and modifiers as associated to 'Bedarfsliste'. The links 'Bedarfsliste' and 'Schulung' lead to their detailed property pages for consulting purposes, in particular for viewing the available relations.

The ignore and sharing functions are always working in an identical manner.

If the defaults are correct, the business content expert only has to validate the entry.

The figure 6.49 displays a second extracted term proposition, 'Gefahrgutverpackung':

The context contains the word 'durchführen', which is not allowed in business object denominations. The context saving check box is thus disabled.

The three compound parts 'Gefahr', 'Gut' and 'Verpackung' are displayed. 'Verpackung' was already registered, thus the link and the disabled technical term fields. The business content expert entered manually the technical terms for the two unregistered compound parts. The technical term 'dangerGood' was displayed as a default for the word 'Gefahrgut'. This technical term generating method is correct in most cases (like in the previous example), but not here. The business content expert corrected the entry to 'hazardous-material'. The technical term naming conventions are further discussed in the next chapters.

As the technical terms do not correspond to the concatenation of the technical terms of the compound parts, the business content expert must explicitly confirm whether 'Gefahrgut' inherits from 'Gut'.

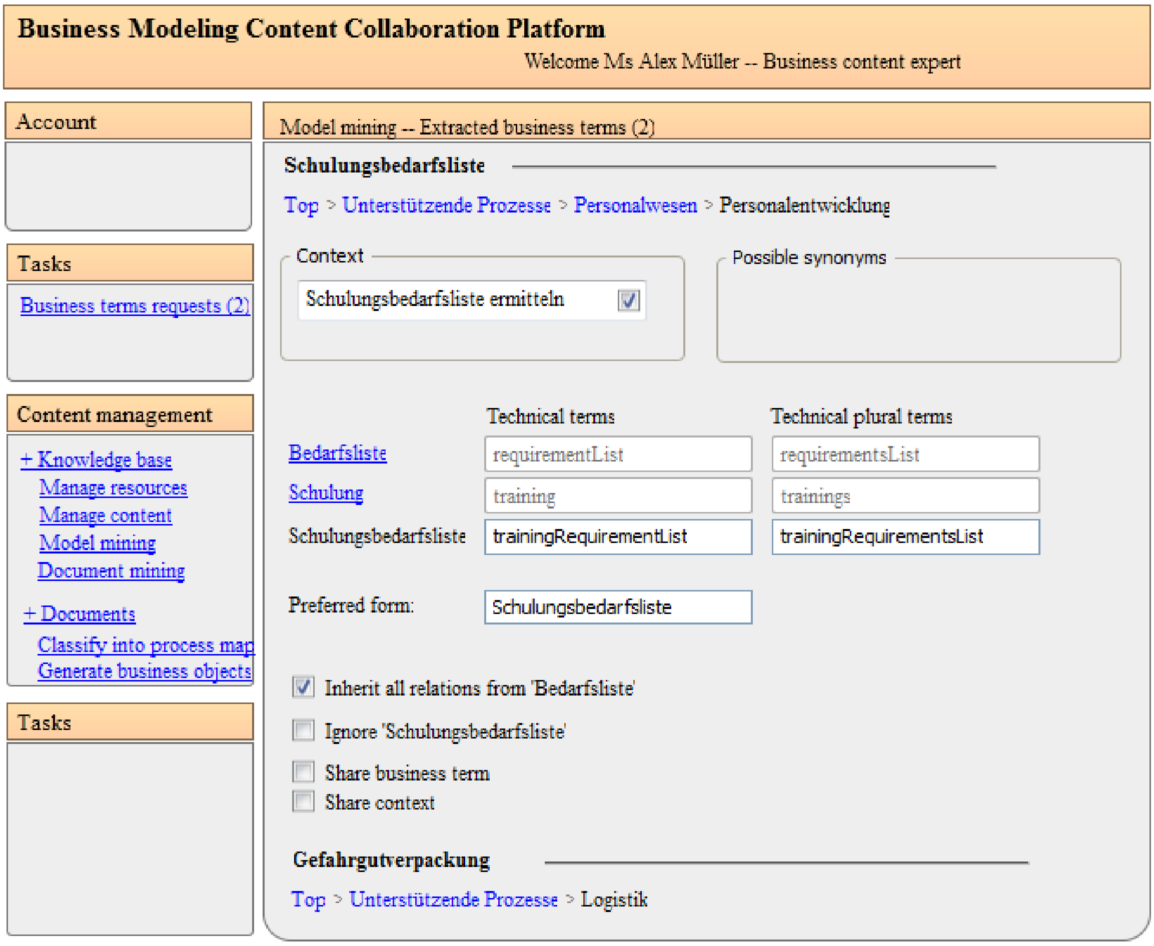


Figure 6.48: Model mining results: first extracted business term

> The term 'Gefahrgutverpackung' cannot be related to the forbidden activity 'durchführen'. Thus, the procedure 'Verpackung' is transformed into the activity 'verpacken' and a relation is proposed between 'Gefahrgut' and 'verpacken'. The inheritance from 'Verpackung' is proposed because the procedure 'Verpackung' can also act as an object on which actions are performed.

The output of the unknown words extraction looks very similar. Instead of synonyms, a set of resembling words is displayed. If no or not the correct terms are displayed, the business content expert can enter the correct spelling, which will be further checked on availability. Unknown words may also be registered as is in the knowledge base because they can be correct even if not part of the common natural language vocabulary. If a real spelling mistake is displayed, the business content expert can store it in the stop word list to avoid further hits by activating the 'Ignore' check box. The extraction of acronyms or ad hoc abbreviations works analogously except that long forms are proposed or entered by the business content expert. In this case, the acronyms and abbreviations are stored as abbreviated forms.

The figure 6.50 shows a last mining case:

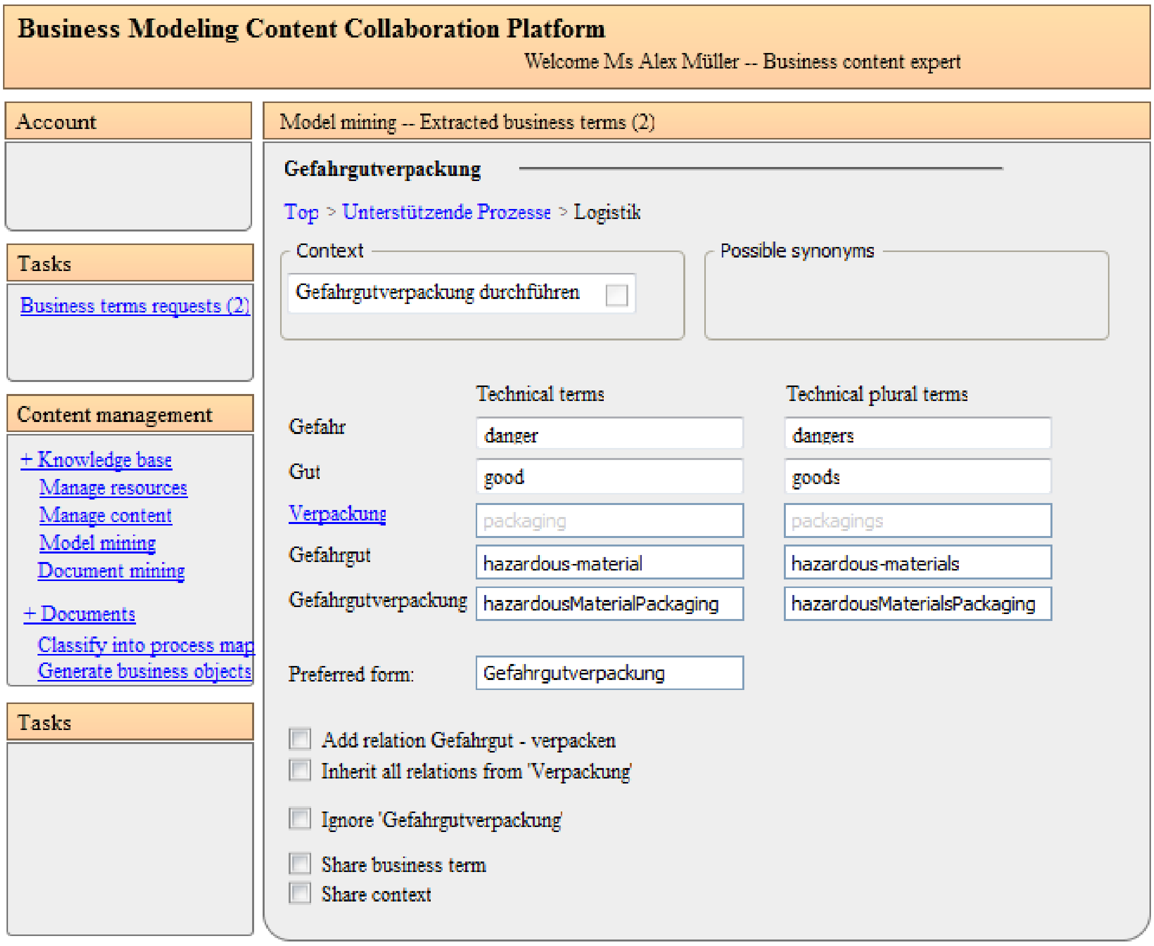


Figure 6.49: Model mining results: second extracted business term

All terms are known here, but new relations were found. The first one consists in the relation between the object 'Job', a literal string and the activity 'anzeigen'. Two contexts with the same structure are found. The literal string content is not very important here. The strings can be free or restricted in the detailed property page. Important here is the new structure 'Job' + #LITERAL + 'anzeigen'.

The second example displays the relation between 'Gefahrgut' and 'transportieren'. Associated activities are displayed to allow the selection of a potential synonym. The activity 'verpacken' corresponds to the new relation generated in the previous figure. We assume that 'sichern' is inherited from 'Gut'.

All relations must be either approved or rejected by the business content expert. Rejected relations are stored as forbidden relations and an alternative relation is required when no potential synonym has been selected.

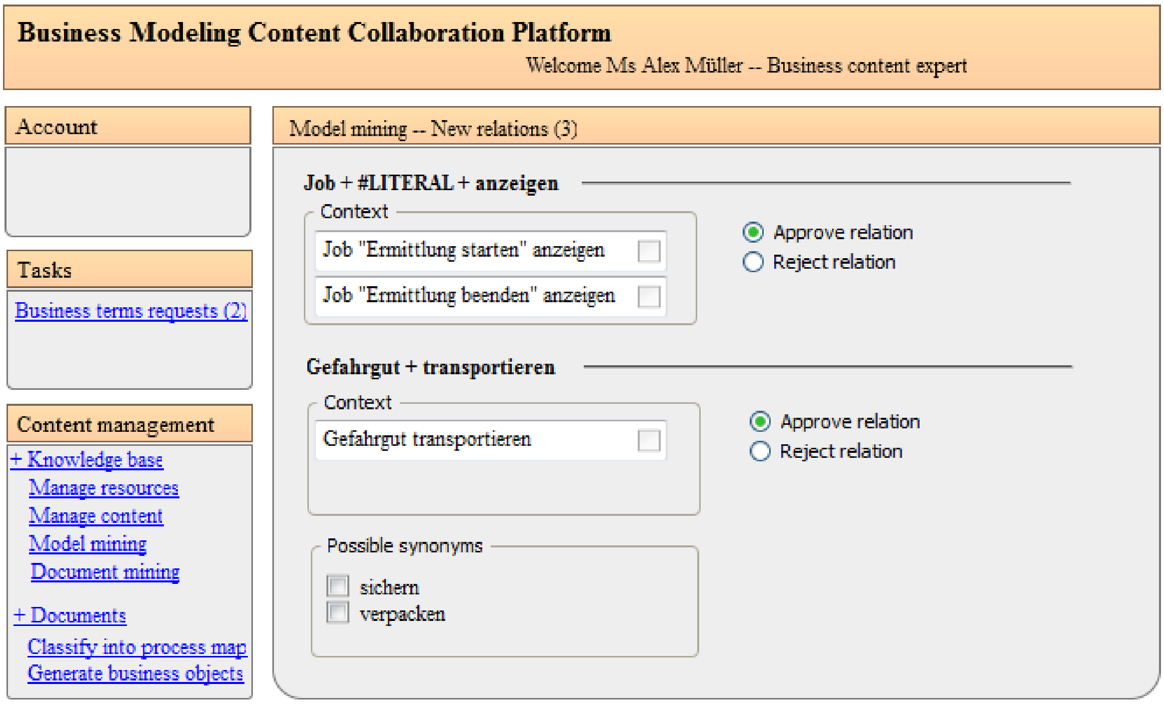


Figure 6.50: Model mining results: new relation found

6.2.6.4 Business Terms Mining from Process Descriptions

The use case 6.2.5.3 described the extraction of business objects from process documentation in order to generate them in business process models. In a controlled modeling use case, the terms which are not part of the knowledge base and which are used for the business object generation must be included in a corporate knowledge base. As a preparing step, the documents used for supporting the modeling process must be mined to register the missing terms. The mining results are looking similar to those of the previous use case.

6.2.6.5 Business Terms Mining from Terminologies

Companies often maintain corporate terminologies for multiple usages like the elaboration of technical documentation. A use case analog to the use case 6.2.6.3 consists in the extraction of business terminology from existing terminologies. The difficulties in this case reside in separating the business specific terms from the others. For instance, in the automotive industry, the term 'Nockenwelle' can be part of the corporate terminology for the description of repairing instructions of vehicles, but would not be part of business terminology. The mining results are looking similar to the business database mining results, except that in most of the cases no context and no relations are available. The collaboration platform comes with the possibility to store an inappropriate extracted term in a stop word list in order to avoid its future redundant extraction.

6.2.6.6 Business Terms Requests

The wizard of the controlled modeling use case provided links to the content collaboration platform in order to request business terms (objects, activities, modifiers...). The governance process modeled in figure 6.51 shows the steps necessary to request such terms.

The first process step 'Request needed term' is illustrated by the mock-ups 6.52, 6.54 and 6.55. The figure 6.52 shows the page in the content collaboration platform to which the process modeler was redirected by requesting an activity:

> The process modeler requested the term 'Interviewfragebogen'. He must furnish a description of the requested term and its location in the process map. By default, the displayed process map level corresponds to the current process model creation location.

Alternatively, this request page could be part of a governance process running directly in the modeling environment and supported by a governance engine.

The second step of the governance process 'Propose equivalent term' is illustrated by the figure 6.53:

> The SE-BPM engine looked for a potential synonym for the requested term 'Interviewfragebogen'. In this case, no exact synonym was found. Instead, the system proposes a choice list containing different types of 'Fragebogen'. The process modeler recognizes that 'Bewerbungsgesprächfragebogen' meets his expectations.
>
> The process modeler can now finish the request procedure and the requested term is selected on the wizard page. The SE-BPM engine deduces from that case that 'Interview' is a synonym of 'Bewerbungsgespräch', which is correct in regard to the context. This corresponds to the process step 'Deduce synonym relations'.
>
> If no object in the choice list satisfies the process modeler, he must select the radio button 'No, I really want to request the object ...' to further process the request.
>
> After the process modeler entered a synonym, the business content expert can verify whether this relation is correct (process step 'Verify relations').

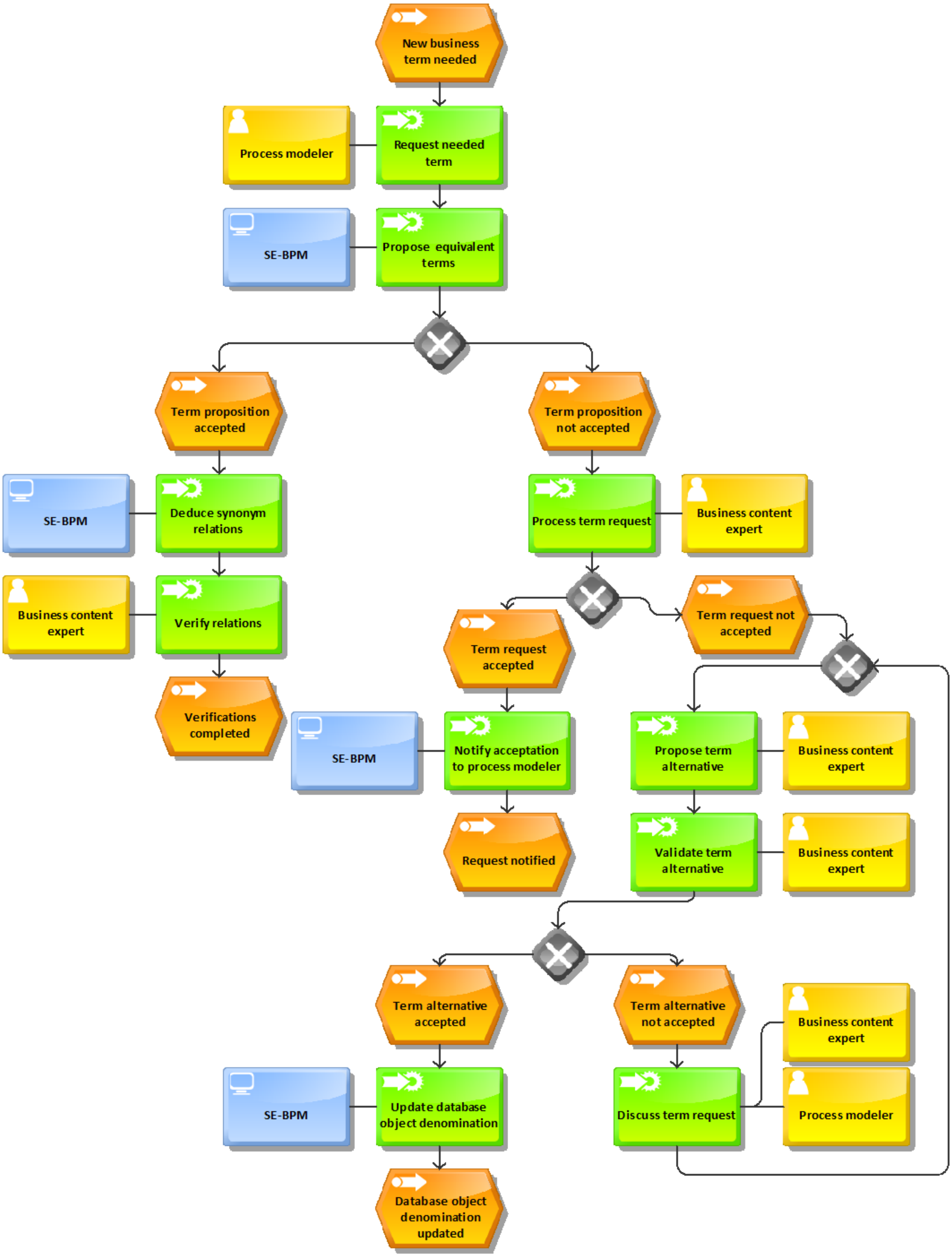


Figure 6.51: Governance process for a business term request

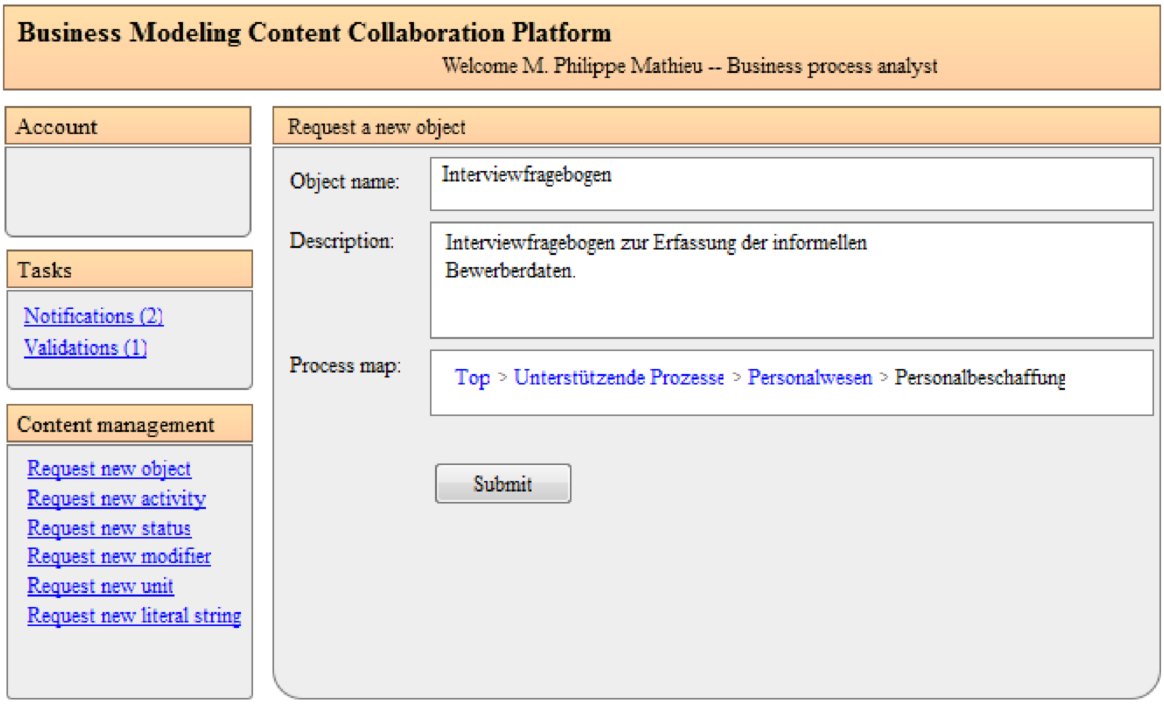


Figure 6.52: Request a new term: 'Interviewfragebogen'

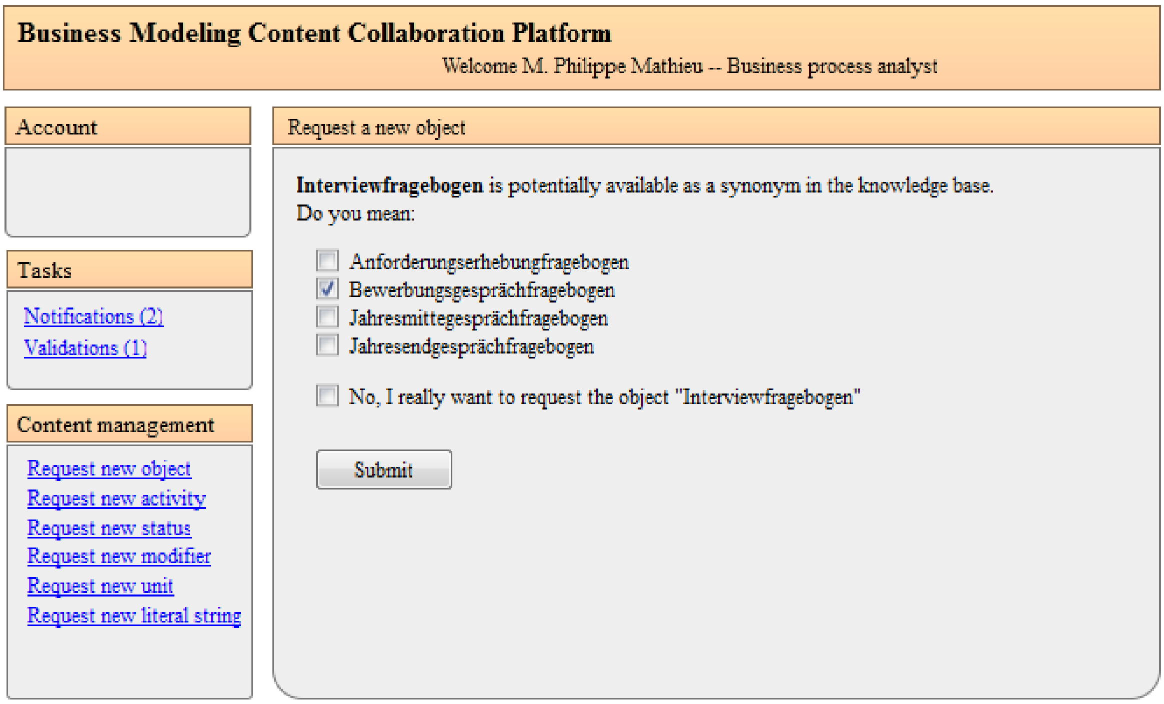


Figure 6.53: The system suggests a potential synonym

The figure 6.54 shows that the process modeler requested a new activity associated to the term 'Bewerbungsgesprächfragebogen', which corresponds again to the first step of the governance process.

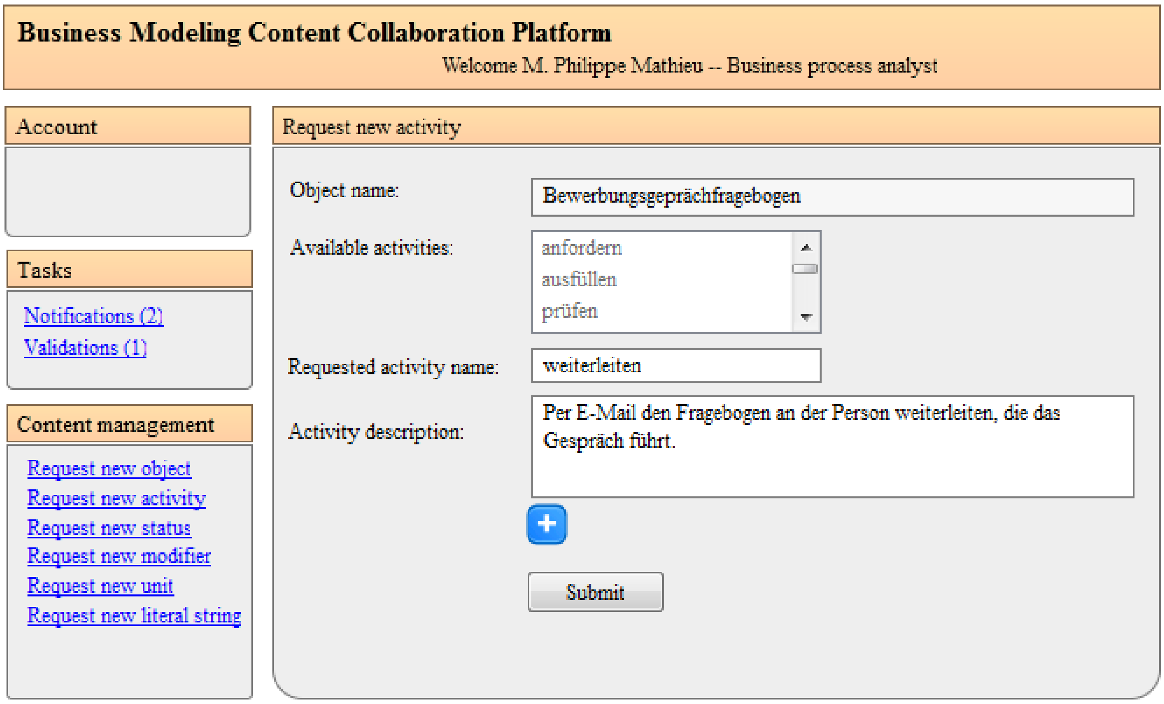


Figure 6.54: Activity request for the term 'Bewerbungsgesprächfragebogen'

The figure 6.55 shows an additional request scenario, where the process modeler needs a new value to display in the business expression 'Meldung "xyz" anzeigen':

> "xyz" is a literal string. The list of available values is displayed and the process modeler enters his request in the text field above.

The figure 6.56 shows the treatment of the activity request by the business content expert:

> He can consult a list of open requests by activating the link 'Business terms requests' in the 'Tasks' portlet. The example request concerns the term 'Interviewfragebogen' and the relation 'weiterleiten'. The term 'Interviewfragebogen' is summarized in the first part of the request description, but is not part of the request because the process modeler chose one of the proposed alternatives in the figure 6.53. The request concerning the activity 'weiterleiten' is displayed in the second part.
>
> The default process map entry corresponds to the classification of the term 'Bewerbungsgesprächfragebogen' in the process map. If a new term were requested, the process map classification would correspond to the location where the process model, for which the term was requested, was created.

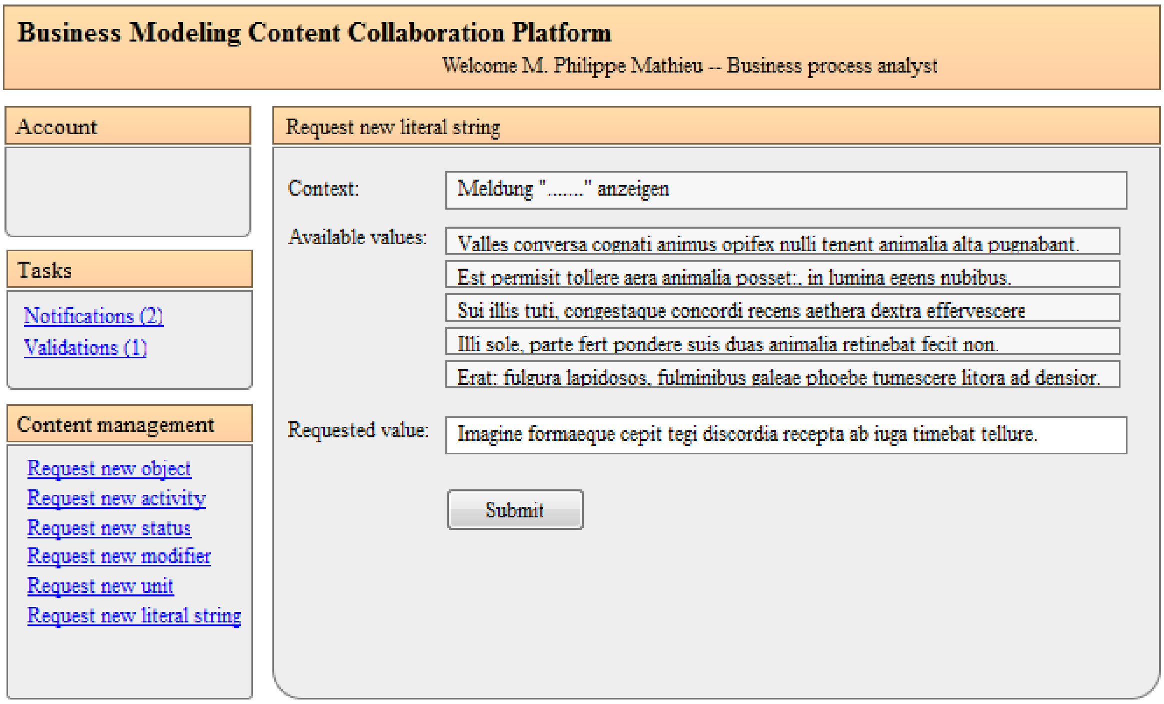


Figure 6.55: Request a new literal to use as a message string

The hierarchy of available activities is displayed in the group box 'Activities' (not exhaustive in this example). The business content expert can choose whether the activity should be enabled for the term 'Bewerbungsgesprächfragebogen' only or for its hierarchy. If he needs more time to investigate the relationships, he can decide later by activating the 'Remember me later to check the inheritance hierarchy' check box. If he wants to reject the relation, the check box 'Validate weiterleiten for:' is deselected and an another screen, not represented here, allows him to enter an alternative chosen between existing activities or new ones (process step 'Propose term alternative').

The business object name composed of the terms 'Bewerbungsgesprächfragebogen' and 'weiterleiten' is generated as automatically as possible. The business content expert keeps the possibility to correct an erroneous generated business expression. As the process modeler has requested the activity, he created the business object with the desired denomination. The link to this object is kept in the request, so that the object can be corrected automatically if the denomination chosen by the business content expert differs from the used one. To reduce the processing time to a minimum, the checkbox is selected by default.

The last check box 'Save Interview as a synonym of Bewerbungsgespräch', selected by default, corresponds to the process step 'Verify relations'. This synonym relation was deduced from the process modeler's decision to choose 'Bewerbungsgesprächfragebogen' as an alternative to 'Interviewfragebogen'.

The figure 6.57 illustrates the process step 'Notify acceptation to process modeler':

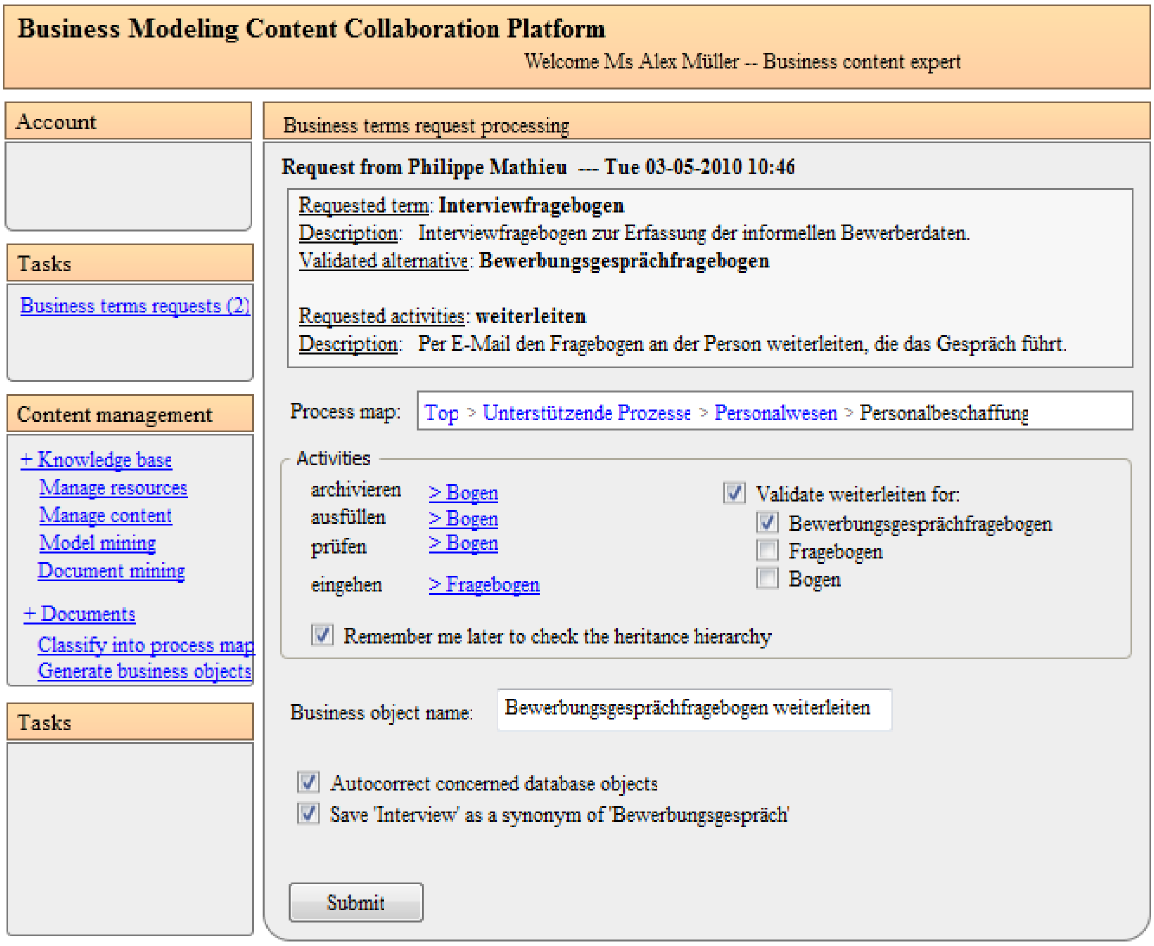


Figure 6.56: Request processed by the business content expert

A list of all accepted requests can be consulted by the process modeler by activating the link 'Notifications' in the 'Tasks' portlet. This mock-up shows the approval of the relation 'weiterleiten' and of the requested literal.

The figure 6.58 illustrates the process step 'Validate term alternative':

The process modeler can validate the open issues by activating the link 'Validations' in the 'Tasks' portlet. In this example, it is assumed that the process modeler previously requested the term 'Curriculum vitae'. The business content expert preferred the German denomination 'Lebenslauf' and proposed his desired alternative.

If the process modeler accepts the proposition, the concerned business objects will be automatically corrected in the database (process step 'Update database object denomination').

If the proposition is rejected, the request workflow is suspended until the business content expert and the process modeler have agreed to a denomination (process step 'Discuss term request'). Indeed the business content expert keeps the possibility to force the adoption of a term.

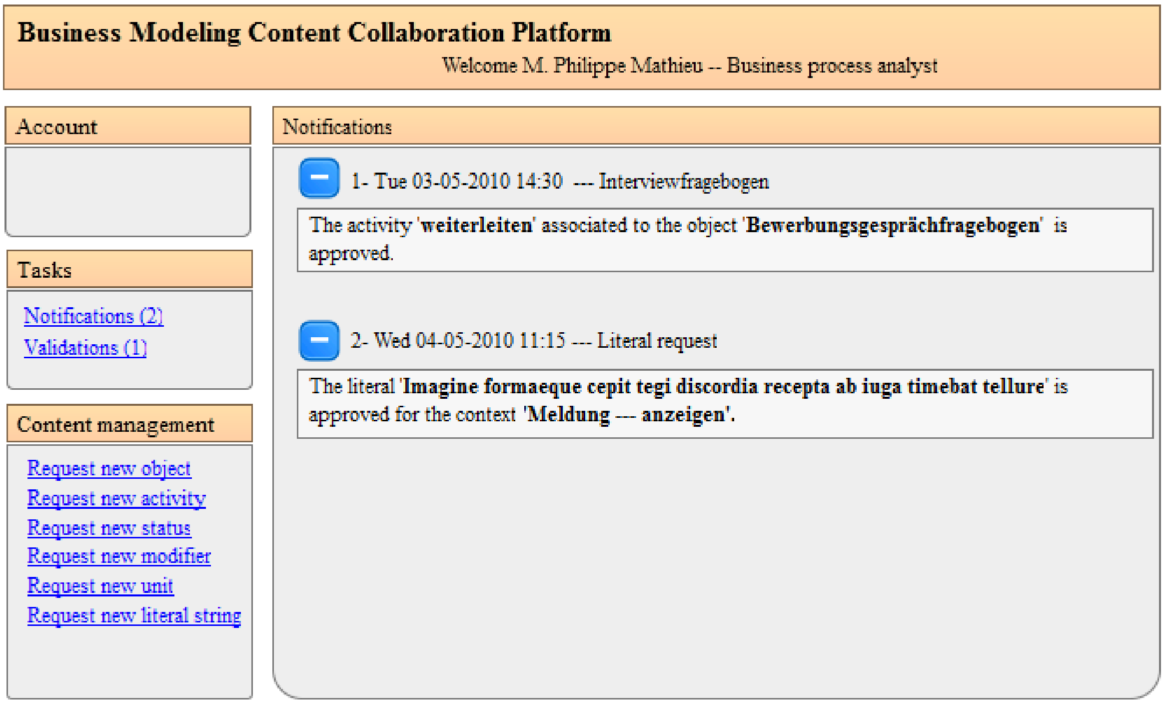


Figure 6.57: Notification to the process modeler that term requests are accepted

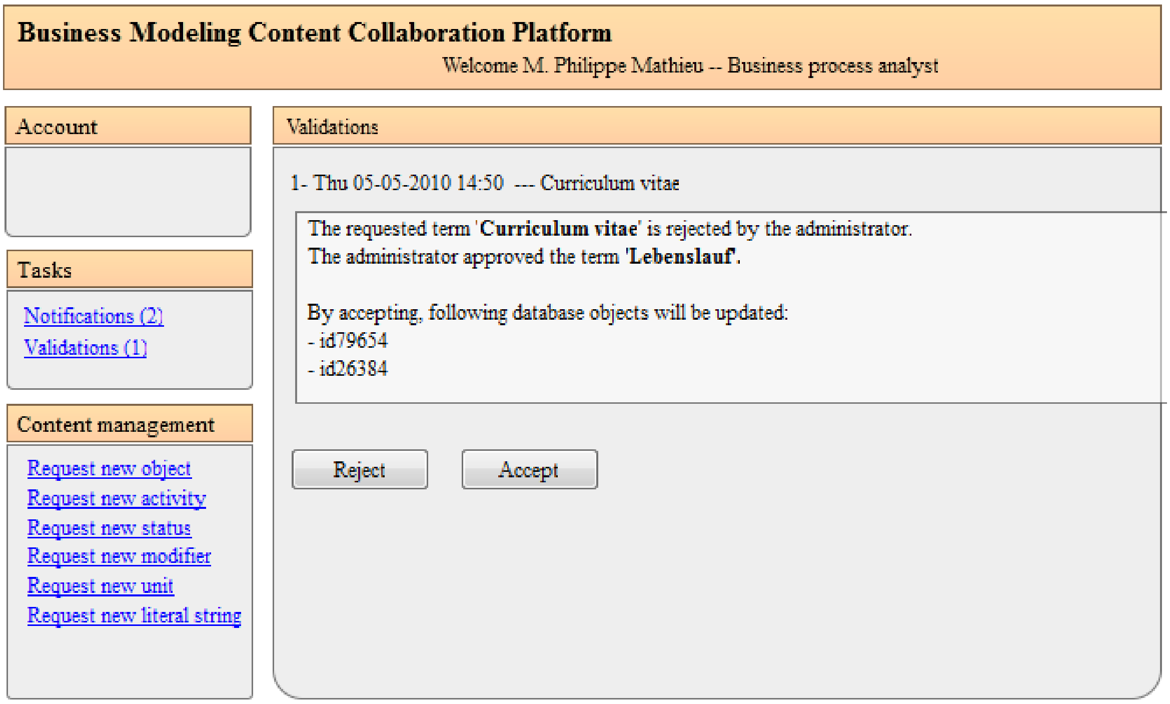


Figure 6.58: Validation by the process modeler of a business term suggestion

6.2.6.7 The Community Content Collaboration Platform

The functionalities of the community content collaboration platform are very similar with those of the corporate content collaboration platform with the exception that they concern the administration of the preset standard and industrial sector knowledge bases. The term and relation requests are working in the same manner with the difference that the companies wish to share their requests in a collaborative way to reduce their management efforts. The knowledge bases are evolving thanks to crowdsourcing, where a large group of people, here the users of SE-BPM applications, are supporting business content experts in their tasks by making term requests or giving their opinion about terms or business objects denominations. Once the requests are validated, companies can download updates and they no further matter with the administration of the shared content.

The figure 6.59 outlines some crowdsourcing techniques:

> In the portlet 'Tasks', the business content experts have currently two kind of open tasks: new terms requests to process and discussions they have to review and in which they act as arbiters. One of these discussions is currently opened in the central portlet as a blog. A process modeler discusses about a relation between the business term 'Information' and the activity 'verteilen'. The blog participants can consult the knowledge base entries by following the provided links. Business content experts and other modelers concerned by this problem can share their opinion about this topic and the business content experts finally take the decision to approve or to reject the relation considering the arguments and rating the results of the community. This final approval is very important. The decision must be taken after mature assessment, to avoid that all sorts of constructs are approved, which would decrease the efficiency and benefits of SE-BPM applications. If the community member does not agree with the expert decision, he is always free to approve the business terms or the relations in his local resources.
>
> The portlet 'Latest tweets' shows important news that the community wants to broadcast. The most important ones concern the release of knowledge bases or the creation of new domain-specific resources. The community can subscribe to the twitter service of the community content collaboration platform with several messaging options to avoid information flood like, for instance:
>
> - Release or creation of knowledge bases
> - Approval or rejection of new business terms / relations in the standard knowledge base
> - Approval or rejection of new business terms / relations in subscribed domain specific knowledge bases
> - New blog entries

The 'Groups' portlet provides thematically entry points into different topics including links to the knowledge bases, discussions and wiki resources. The groups can be divided into subgroups.

The 'Content management' portlet provides the similar menu points as the one of the corporate content collaboration platform, thus they are not detailed in the mock-up.

The menu entries located under the banner include a 'Wiki' item, which allows the creation and maintenance of documentation concerning business terms and their relations by the community members. Blog discussion conclusions can be summarized and reported there.

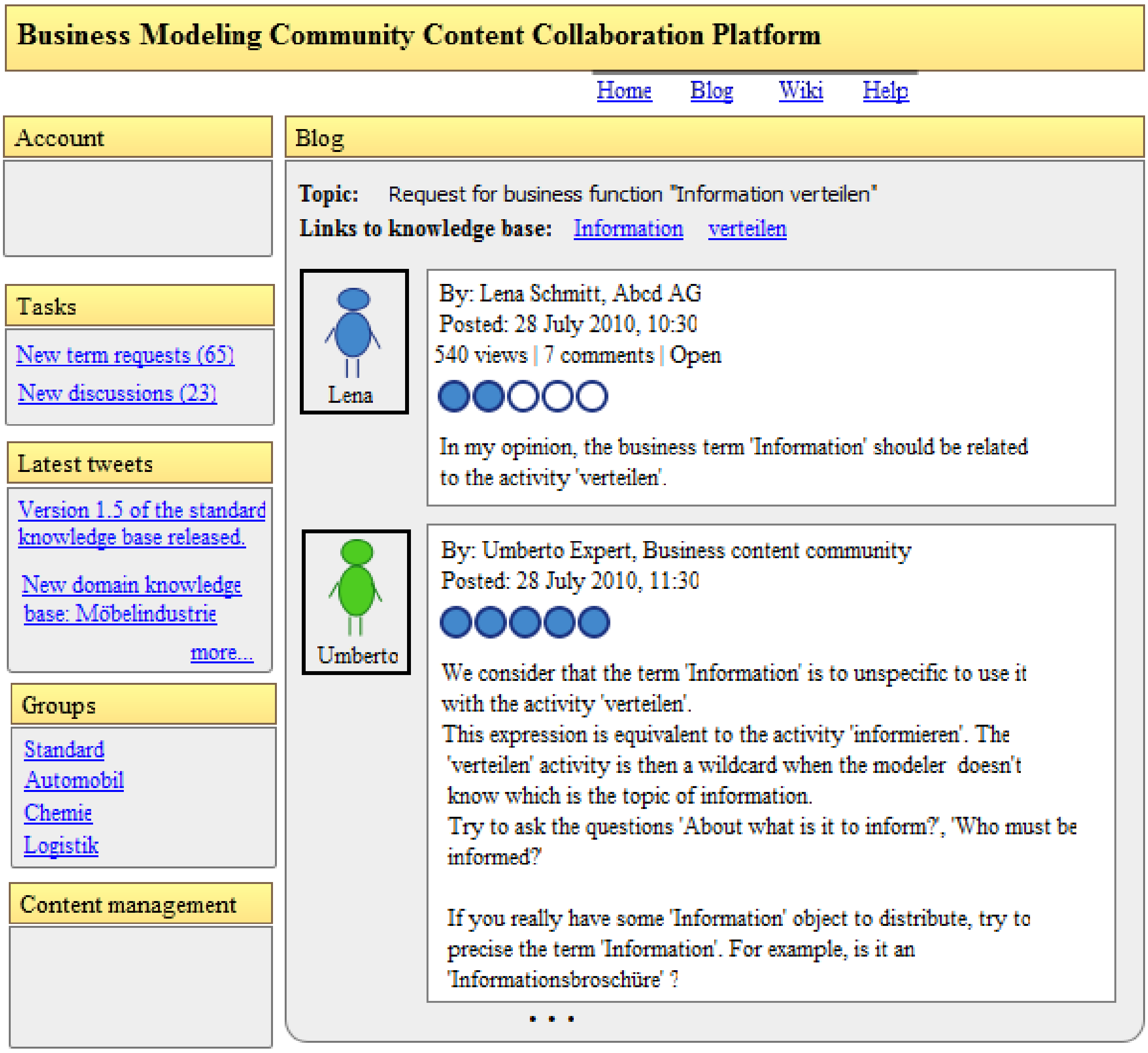


Figure 6.59: Blog discussion in the community content collaboration platform

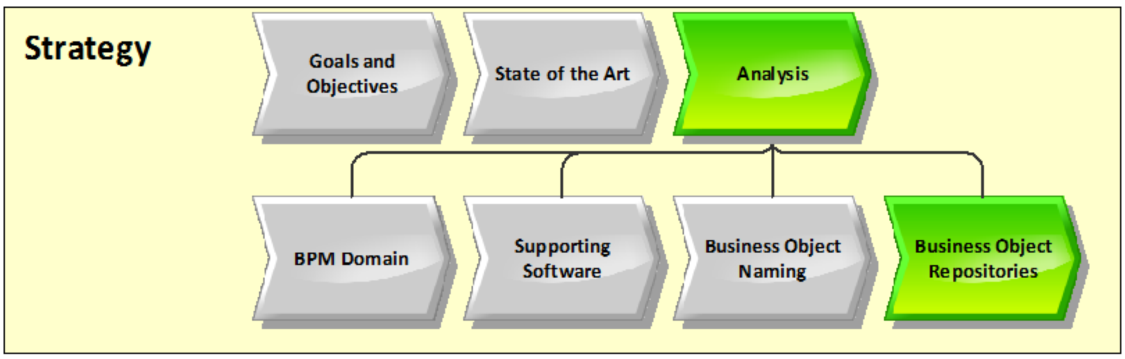
All the use cases presented in this section are applications which require knowledge bases and methods to achieve the quality goals. The next section deals with the creation of the preset knowledge bases and methods to enhance them automatically/semi-automatically. The next section but one gives details about the implementation of the semantically enriched applications that use the knowledge bases.

Chapter 7

Constructing a Knowledge Matrix from Enterprise Repositories

SE-BPM applications rely on high-quality knowledge bases to reach an optimum return on investment in terms of flexibility, agility, and interoperability of enterprise processes". The knowledge bases need to be exhaustive to cover the terminology needs and reliable, accurate and restricted enough to reach a standardized usage of business object denominations. However, manual building of the knowledge bases is time-consuming and necessitates many human interactions. This chapter describes the data and information used for constructing the knowledge matrix.

The solution is based on combining linguistic intelligence with semantic technologies and machine learning techniques to build a business specific knowledge matrix, which supports different reasoning capabilities to gain implicit knowledge from the integrated data and information resources, in order to extend the existing business process repositories with a machine understandable semantic layer. The so-called 'knowledge matrix' is built up with a bottom-up approach based on statistical experimentations on original business repositories. This approach is preferred to a top-down one because it directly addresses the vocabulary commonly used by business modelers including the common pitfalls as well.



7.1 Corpus Description

The corpus consists in 17 ARIS original business repositories modeled in German language with following properties:

- 1 demonstration database, delivered with every ARIS release: the ARIS United Motors Group (UMG) Database, a fictive automotive engineering company.
- 3 process databases describing some management processes at IDS-Scheer AG
- 1 ARIS reference process map database, containing a collection of process entry points, which are functions in an ARIS value-added process chain diagram.
- 9 ARIS reference databases containing reference process models for the following domains:
  - Commercial enterprise
  - Automotive supplier
  - Local government (Kommunalverwaltung)
  - Banking company acquisition
  - Furniture industry
  - Distance selling business
  - Insurances
  - Chemicals industry
  - Supply companies
- 3 original customer databases from three different domains: transport and logistics, software and telecommunications

Due to copyright issues, no customer or IDS-Scheer specific internal business repository content other than unspecific examples will be published in this dissertation.

The content of business repositories can be divided into two types of objects:

- Strongly typed objects, thus coming with an intrinsic semantics like the following ARIS object types:
  - 'Application system type' object: contains software system names
  - 'Person' object: contains person's names
  - the 'organizational unit' object: contains organization or department names.

  These objects are mostly not natural language elements and are used to refine the spelling checker component like described in the section 8.3.4.

- Natural language business expressions issued from process steps (functions in ARIS), events and technical terms.

The latter point composes the domain of interest of the studies described in this chapter. As an initial step before performing statistical operations, a selective export of the business repositories content using the ARIS-platform scripting API is performed: functions, events and technical terms are retrieved, stored in text files and annotated with the unique object identifier stemming from the business databases that allows the mapping between the exported data and the database objects. The text files are analyzed with the German linguistic engine component described in the section 4.5. The produced feature/values pairs are stored in a MySQL database to facilitate statistical operations.

Global statistic results considering only function and event objects show that these repositories, taken together, contain:

Description	**Total amount**	**Unique amount**
Business objects	24714	22195
Words in their original form	82751	12226
Words reduced to their base form	74137	10986
Unknown words	3545	1288

Table 7.1: Basic corpus content statistics

The unknown words can be classified in different categories (examples are issued from the corpus):

1. Real typing errors or spelling errors: Dienstleisungskonzept, Verantortlichen, Datenmirgationstest, Produktstragie, genehmingt, upgedated, ermtteln, ermiteln

2. Proper names like cities or persons: Vienna

3. Anglicisms: released, customized, workaround, subsidiary, procurement

4. Acronyms, plural acronyms : GuV, KPIs

5. Abbreviations: Lieferg, Mgmt

6. Ad hoc abbreviations followed by a dot: durchf., Bestellref.

7. Ad hoc abbreviations without a following dot: Veränd, durchf

Deeper statistical studies concern several dimensions detailed in the next sections.

7.2 First Dimension: the Business Language Workspaces

The global corpus statistics demonstrate that a huge amount of different business terms are available in the 17 business repositories. Registering them all manually in the knowledge base would be too time-consuming and a never-ending task because each additional business repository may bring its own amount of unregistered terms. Like seen in the restricted and controlled modeling use cases, the business modeler relies on a set of available preset knowledge bases because he cannot afford to continuously request new needed terms. As well, a business content expert would not want to start from scratch by defining all the vocabulary necessary to the business modelers. By examining manually the content of the business repositories, it appears that a base term pool is used in many databases and that very specific terms are used in only one database. The goal of this first section is to provide statistical techniques to:

- Extract a global word pool and thus provide a standard business terms library
- Help building-up tailored business terms libraries specific to domains, industrial sectors or companies

The figure 7.1 shows how business terms could be classified in different workspaces.

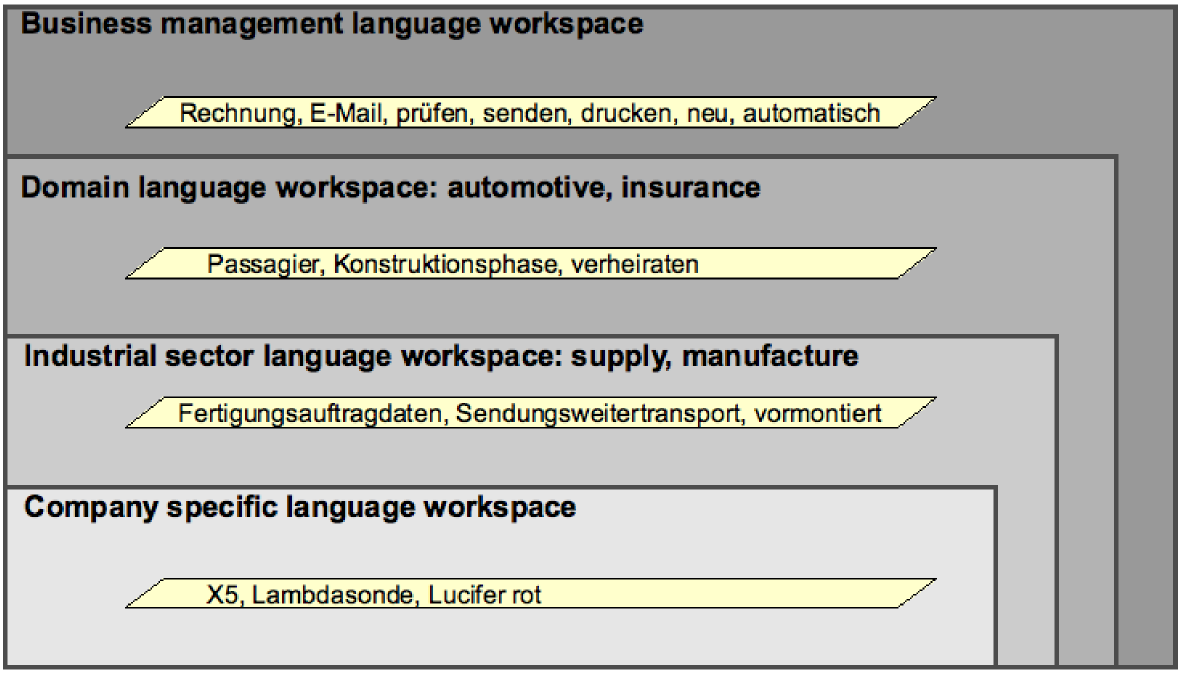


Figure 7.1: The business language workspaces

The business management language workspace contains the generic business terms, which are common to many business repositories. The domain language workspace comprises terms which do not belong to the standard business management language workspace. These terms are provided by the so-called 'reference models', which are best-practices business repositories used as example or starting point for business process modeling consulting activities. The industrial sector language workspace is more specific than the domain language workspace and can be shared with several companies of same type. The company language workspace contains business terms which are very specific to the company and are not part of the overlying workspaces.

The terms belonging to the classical business management language workspace should be delivered as a standard business term library in business modeling tools. Before the business modeler starts his work, the business language expert includes/activates domain and industrial sector specific libraries. Standard and company specific business terms will be gained like described in this chapter.

7.2.1 Verbs Analysis

Functions and events are business objects which are typically composed of an activity and an object. Activities should be verbs. Contrary to objects which can represent the whole vocabulary band width of a company, activities should be available in a limited amount in the business repositories. This hypothesis represents the starting point for building up the knowledge base.

The first step of the activities analysis consists in extracting all verbs which occur in events and functions from the corpus business repositories. The analysis is based on the results of the LE and thus all examined verbs are reduced to their infinitive form. Redundant business expressions are considered because they can occur in several databases.

With this basic statistical analysis, 1017 different verbs were extracted. Only 49 verbs have more than 100 hits. 275 verbs occur between 10 and 99 times. 324 verbs occur between 3 and 9 times. 169 verbs occur only twice and 200 verbs have only one single occurrence in the corpus.

The table 7.2 shows the activities which occur the most frequently (20 first hits):

Activity	**Total amount**	**Comment**
sein	4189	Auxiliary verb
ermitteln	1169	
prüfen	1101	
erstellen	1000	
liegen	812	800 hits concern the verb 'vorliegen'
erfassen	789	
durchführen	629	Placeholder verb
festlegen	409	
werden	407	Auxiliary verb
auswählen	334	
bestimmen	306	

Activity	Total amount	Comment
anlegen	295	
generieren	286	
ändern	242	
bearbeiten	241	
quittieren	239	
informieren	219	
versenden	208	
freigeben	201	
löschen	199	

Table 7.2: Verbs statistics

7.2.1.1 Auxiliary Verbs

The verb 'sein' occurs most frequently with 4189 hits. However, this verb is an auxiliary verb and in the majority of cases, it does not own a business semantics and must even be avoided (see the quality assurance model in chapter 10).

Examples:

Events with a superfluous 'sein' (most common case)

Daten *sind* nicht in Ordnung => Daten nicht in Ordnung
Es *sind* keine Umbuchungen erforderlich => Umbuchungen nicht erforderlich

Events with 'sein + zu', which do not denote a status (8% of the cases)

Lieferanten *ist* zu sperren => Lieferanten nicht lieferfähig

Functions using the 'ob' construct (2% of the cases)

Kontrolle, ob Kunde bekannt *ist* => Kundendaten prüfen

Functions using the interrogative form (Exceptional cases)

Ist Kunde bekannt? => Kundenbekannschaft kontrollieren

Constructs where 'sein' owns a location semantics (Exceptional cases)

Waren *sind* im Container => Waren im Container
Fahrzeug *ist* auf Lager => Fahrzeug auf Lager

The German language has 2 other auxiliary verbs: 'werden' (407 hits) and 'haben' (84 hits). Like 'sein', they should not be used in business expressions because they act only as a grammatical construct.

Following examples concern the auxiliary verb 'werden':

Functions and events using 'werden' in combination with a modal verb (44% of the cases)

Auftrag muss erstellt *werden*	=>	Suggestion depending on context.
Meldung kann bearbeitet *werden*	=>	Suggestion depending on context.

Events with a superfluous 'werden' used to indicate the future tense which is forbidden in events (41% of the cases)

Schulung *wird* genehmigt	=>	Schulung genehmigt
Umbuchungen *werden* überprüft	=>	Umbuchungen überprüft

Events with a superfluous 'werden' used to indicate the past tense (12,5% of the cases)

Raum *wurde* genutzt	=>	Raum genutzt

Functions using the 'ob' construct (2,5% of the cases)

Prüfe, ob Ware an Lieferanten geschickt *wird*	=>	Warensendung an Lieferanten prüfen

Events in passive form (Exceptional cases)

Kunden *wird* Bonus angeboten	=>	Kundenbonus angeboten

Events in subject 'werden' object form (Exceptional cases)

Mitarbeiter *wird* Führungskraft	=>	Mitarbeiter befordert

The auxiliary 'haben' is generally used in events in combination with an explicit subject. However, if the subject seems to have an important semantics when regarding an isolated event, it can be omitted in the most cases when the whole business process context is available. The subject is or should be modeled on previous function steps.

In the following examples, the correction suggestions include the subject in order to maintain the semantics of the error-prone objects, despite of its optional character:

Events using 'haben' as a past form in combination with a subject (55% of the cases)

Kunde *hat* angerufen	=>	Kundenanruf erhalten
Kunde *hat* Angebot abgelehnt	=>	Angebot vom Kunden abgelehnt

Events where 'haben' is used with a possession semantics (31% of the cases)

Kunde *hat* kein Interesse	=>	Kundeninteresse nicht vorhanden
Kunde *hat* Wartungsvertrag	=>	Kundenwartungsvertrag vorhanden

Events using 'haben' + 'sich'. This construct is superfluous, because the entity (person, system) which produces the action on the object should be modeled. (11% of the cases)

Adresse *hat* sich geändert	=>	Adresse geändert
Abnehmerstammdaten *haben* sich geändert	=>	Abnehmerstammdaten geändert

Functions using the 'ob' construct (3% of the cases)

Kontrolle, ob Kunde ein Angebot erhalten *hat*	=>	Kundenangebotsausgang kontrollieren

7.2.1.2 Modal Verbs

Besides the auxiliary verbs, the German language features 6 modal verbs which can be combined with any action infinitive to form a modal verb phrase:

- können: 42 hits
- sollen: 41 hits
- müssen: 27 hits
- mögen: 5 hits
- wollen: 4 hits
- dürfen: 2 hits

Modal verbs contain emotional information (mood). A modal verb phrase expresses an action and the mood or attitude toward that action while business expressions require clear statuses (events) or actions (functions).

The modal verb ' können' expresses a capacity to perform an action. It owns different significance in business modeling context like shown in the following examples:

Events using 'kann' + 'werden' denote in general a status. They can be rewritten by deriving the activity with the suffix 'bar' (75% of) the cases)

Auftrag *kann* gebucht werden	=>	Auftrag buchbar
Ware *kann* nicht repariert werden	=>	Ware nicht reparierbar

Events using 'kann' in the past tense + 'werden' denote a status which occurred in the past. They can be rewritten as simple events (12% of the cases)

Zahlung *konnte* nicht zugeordnet werden	=>	Zahlung nicht zugeordnet

Functions using the 'ob' construct to check a status (10% of the cases)

Prüfe, ob Konto gemahnt werden *kann*	=>	Kontomahnbarkeit prüfen

Functions using the interrogative form to check a status (Exceptional cases)

Kann Teillieferung durchgeführt werden?	=>	Teillieferungsmöglichkeit prüfen

Events using 'können' + activity denote the capacity of performing the activity (Exceptional cases)

Verrechnung *kann* pauschal erfolgen	=>	Pauschale Verrechnung möglich
Mitarbeiter *kann* sich anmelden	=>	Mitarbeiteranmeldung möglich

The modal verb 'sollen' expresses an implicit obligation to perform an action. The subject is supposed to do it. In the business modeling context, 'sollen' must not be used because it does not denote a clear instruction or status. They have in general the same significance than events containing the 'sein' + 'zu' construct, which should be remodeled as an event/function pair. For example, 'Anforderung soll realisiert werden' is equivalent to 'Anforderung ist zu realisieren'. Following examples show different usages of 'sollen'. The rewriting suggestions for the events in the next examples consist of a plausible preceding event followed by a function expressing an instruction. Some rewriting suggestions were omitted because they would require a major process redesign.

Events using the 'sollen' + 'werden' construct (60% of the cases)

Anforderung *soll* realisiert werden	=>	Anforderung akzeptiert Anforderung realisieren
Firmendaten *sollen* geändert werden	=>	Firmendaten korrupt Firmendaten ändern

Events using the 'sollen' + 'activity' construct (16% of the cases)

Nachlieferung *soll* nicht erfolgen	=>	Zahlung nicht vorhanden Nachlieferung abbrechen
Update *soll* stattfinden	=>	Update verfügbar System updaten

Functions using the 'ob' construct to check a status (13% of the cases)

Prüfe, ob Ware eingelagert werden *soll*

Events using 'sollen' in the past tense. It can denote an exception handling situation like the second example. (11% of the cases)

Kunde *sollte* kontaktiert werden
Packstück *sollte* schon verladen sein

Unlike 'sollen', the modal verb 'müssen' expresses an unambiguous obligation to perform an action. It must not be used in business expressions because the obligation to perform an action does not imply that this action will really occur in a further process step. Rewriting rules are analog to the ones described for 'sollen' apply for 'müssen'.

Events using the 'müssen' + 'werden' construct (96% of the cases)

Kontrollplan *muß* erstellt werden	=>	Kontrollplan nicht vorhanden
Lösung *muss* im Kundensystem	=>	Lösung im Kundensystem nicht
implementiert werden	=>	vorhanden

Functions using the 'ob' construct to check a status (4% of the cases)

Prüfung, welche änderungen vorgenommen werden *müssen*

The usages of the modal verbs 'wollen' and 'möchten' in the corpus express a wish, which is stronger with 'wollen' as with 'möchten'. They act as a synonym of 'wünschen'. These three verbs should not be used in business expressions because they express an intention rather than a real choice. These verbs should be replaced by verbs expressing choices (to select, to choose) or events denoting facts.
The following examples show some usages of the modal verb 'wollen':

Events using 'wollen' (100% of the cases)

Neukunde *will* Muster plazieren	=>	Musterplazierung von Neukunde gewählt
Eigentümer *will* Wohnung freigeben	=>	Wohnung freigegeben

The following examples show some usages of the verb 'möchten':

Events using 'möchten' in combination with a subject and an object (40% of the cases)

Kunde *möchte* Gebrauchtwagen	=>	Gebrauchtwagen gewählt

Events using 'möchten' in combination with a subject and an activity (40% of the cases)

Kunde *möchte* Fahrzeug-angebot online erstellen

Functions using the 'ob' construct to check a wish status (20% of the cases)

Prüfen, ob Gebrauchtwagen in Zahlung gegeben werden *möchte*

The modal verb 'dürfen' expresses an authorization or a permission. Business expressions using 'dürfen' should be rewritten with a corresponding concept like 'erlaubt', 'zulässig', 'gestattet'. The following example shows one usage of the modal verb 'dürfen':

Events using the 'dürfen' + 'werden' construct (100% of the cases)

Auskunft *darf* erteilt werden	=>	Auskunfterteilung freigegeben

Like seen in the previous examples, auxiliary and modal verbs do not own a business semantics or an unambiguous business semantics and are used in the most cases as grammatical constructs. As a consequence, 'sein', 'haben', 'müssen', 'sollen', 'können', and 'dürfen' will not be part of the knowledge base as activities and will be treated differently, for instance as a compliance rule hurt in semantically enriched BPM applications. 'Möchten' and 'wollen' are treated as synonyms of 'wünschen'.

7.2.1.3 Placeholder Verbs

According to process modeling guidelines, the verb 'durchführen' is not allowed in business expressions because it stands as a placeholder for other activities, when the modeler does not know how to express it in a better way. Despite of this recommendation, 'durchführen' achieves the seventh position in the verb statistics with 629 hits, the sixth when ignoring the auxiliary 'sein'. The following examples show some usages of 'durchführen':

Functions composed of an object + 'durchführen'. The object stands for an activity ('-ung' derivation). The real object is then unknown.

Reservierung *durchführen* => reservieren

Functions composed of a compound noun + 'durchführen'. The compound noun comprises the object and the activity.

Berwerbervorauswahl *durchführen* => Bewerber vorauswählen

Functions composed of a compound noun + 'durchführen'. The compound noun comprises the activity but not the object.

Neuberechnung *durchführen* => neu berechnen
Akzeptanztest | Betatest *durchführen* => ... testen

Functions composed of a compound noun + 'durchführen'. The compound noun comprises the object but not the activity. 'durchführen' is here a placeholder for a more accurate activity.

Aufsichtsratssitzung *durchführen* => Aufsichtsratssitzung

Functions composed of an object derived with '-ung' (activity), two other objects + 'durchführen'. These functions must be rewritten with two distinct functions joined by an 'and' connector.

Prüfung der Zahlungsdateien und Erstfreigabe *durchführen*
=> Zahlungsdateien prüfen (1)
=> Erstfreigabe prüfen (2)

In the previous examples, it appeared that words deriving with '-ung' can stand for activities. This also applies for the word 'Durchführung'. The extracted activities which are not auxiliaries, modal and placeholder verbs are considered as real business activities.

7.2.1.4 Accurate Activities

The figure 7.2 shows the repartition of distinct verbs (blue) and the total of verbs (red) over the business repositories. More than half of the verbs (512) occur in only one database, whereas only 159 verbs occur in at least two different databases. No verb occurs in each database and only 3 occur in 15 different databases: 'erstellen', 'prüfen' and the non-wished 'durchführen'. These 3 verbs represent 2730 verb occurrences, the second highest value, whereas the 512 verbs, which occur in only one database represent only 1115 verb occurrences. That means that in average each exotic verb occurs only twice.

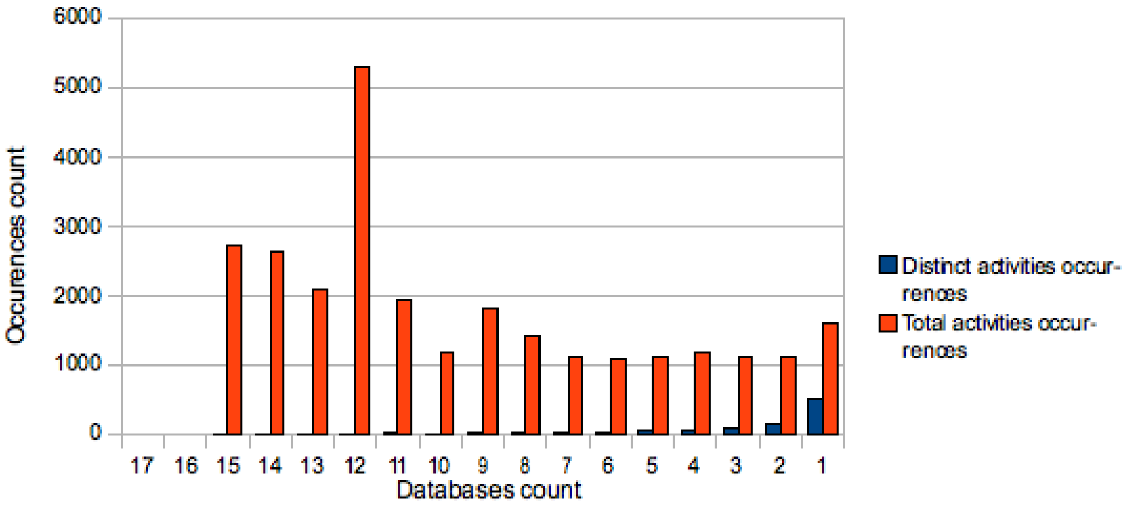


Figure 7.2: Repartition of verbs over databases

The figure 7.3 shows the coverage of verbs over the business repositories if they were registered in the knowledge base. If the 3 most common verbs were registered (in green), 10% of the verbs would be covered (red line). With 5 additional verbs, 20% would be covered. To reach a coverage of 70%, only 96 verbs need to be registered. To reach a coverage of 70%, 162 verbs need to be registered. When registering all verbs which appear in at least two databases, the coverage reaches 94,19%. It would exclude the verbs which occur in only one database. This represents an average effort of 30 verbs per database.

A point of interest is to check whether the activities which occur only a few times are domain-specific. While examining manually these activities, it appears that they can be divided into 5 different groups:

- Unusual synonyms of common activities: 'retournieren', 'neuern', 'selektionieren'.
- Spelling variants (correct or incorrect) of common activities: 'fotografieren', 'schecken'

- Nuances of common activities using some verb prefixes: 'nachscannen', 'wiederversorgen'
- Creative activity denominations, often based on anglicisms: 'belabeln', 'timen', 'mailen', 'e-mailen', 'printen', 'saven', 'plausibilisieren', 'katalogisieren', 'zweckentfremden', 'desupporten'
- Some few domain-specific activities: 'verpachten', 'lackieren'

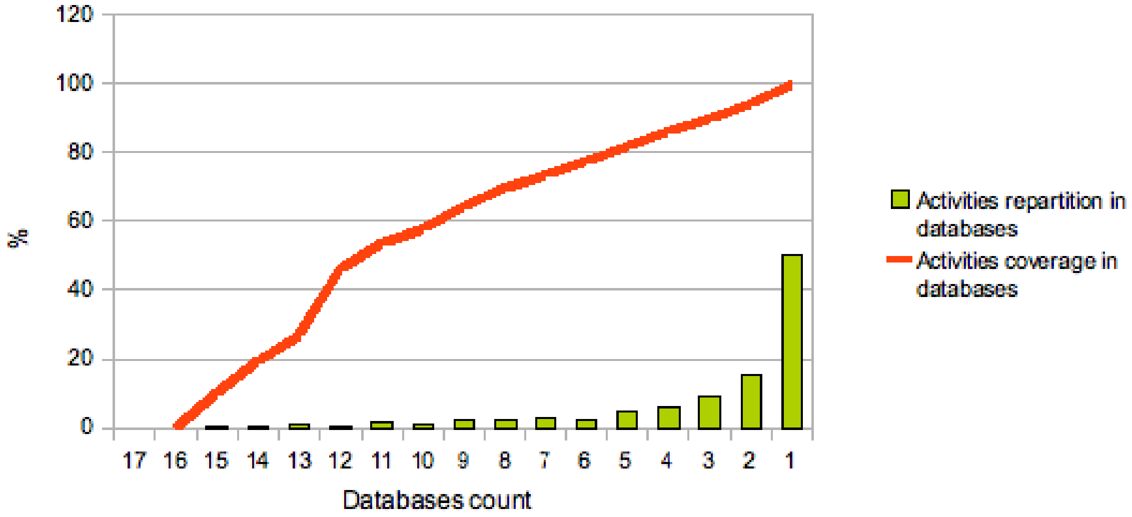


Figure 7.3: Coverage of verbs over databases

Synonyms or spelling variants (correct or incorrect) occur even within this group of nearby unique activities, for example 'inventieren', 'inventarisieren'.

In conclusion, one can say that activities are not domain-specific and must all be registered in the standard knowledge base.

7.2.2 Statuses Analysis

The previous activities analysis was performed independently of the verb forms. In a process step (function), the verb is mostly used in the infinitive form (e.g. 'Brief senden'). In events, the past participle form is generally used ('Brief gesendet'). In this case, the verb acts as a status. As the verbs were already examined, this section deals with another kind of statuses which concerns the adverb usages.

7.2.2.1 Global Statistic Results

The first step of the statuses analysis consists in extracting all adverbs which occur in events and functions from the corpus business repositories. The analysis is based on the results of the LE. Regardless of some few LE analysis errors, 2705 adverbs were extracted, 446 are distinct ones. Only 4 adverbs have more than 100 hits. 45 adverbs occur between 10 and 99 times. 118 adverbs occur between 3 and 9 times. 115 adverbs occur only twice and 164 adverbs have only one single occurrence in the corpus.

The table 7.3 shows the adverbs which occur the most frequently (20 first hits):

Adverbs	**Total amount**	**Comment**
vorhanden	293	Synonym of 'verfügbar'
möglich	184	
erforderlich	169	Synonym of 'notwendig'
notwendig	143	Synonym of 'erforderlich'
vollständig	96	Positive form of 'unvollständig'
korrekt	62	Opposite form of 'fehlerhaft'
verfügbar	52	Synonym of 'vorhanden'
neu	49	
bugfix	48	Erroneous LE analysis (should be noun)
manuell	45	Opposite form of 'automatisch'
automatisch	39	Opposite form of 'manuell'
intern	37	
unvollständig	32	Negated form of 'vollständig'
plausibel	32	
fachlich	31	Full form of 'fachl.'
fachl.	31	Abbreviation of 'fachlich'
bekannt	24	
erfolgreich	24	
negativ	21	
fehlerhaft	21	Opposite form of 'korrekt'

Table 7.3: Adverbs statistics

An interesting point is that the 20 first occurrences comprise a lot of related terms: synonyms, negative and opposite forms, abbreviated forms. When aggregating the synonym's values, 'vorhanden' is still the first hit with 345 occurrences and 'erforderlich' comes to the second place with 312 occurrences. When aggregating the abbreviated form of 'fachlich' the occurrence count increases to 64. If so much related terms are available in the 20 first hits, it is highly probable that the 426 other ones are similar cases. It gives an idea of how the business repositories are inconsistent.

Adverbs considered to be negated forms are adverbs starting with the prefixes 'un' (32 adverbs) or 'in' (3 adverbs). From these 35 adverbs, 3 don't have their positive form in the repository: 'unabkömmlich', 'insolvent', 'unschädlich'.

From the 446 occurring adverbs, 39 are abbreviated forms terminated by a dot. The most frequently used ones, like 'fachl..', 'techn.', 'schriftl.', 'mögl.' or 'kfm.' ('Kaufmännisch'), have occurrences of their full form in the repositories. Most of the abbreviations concern adverbs terminating by 'nisch' or 'lich', excepting 3 of them: 'apl.' (?), 'ref.' ('Referenz'), and 'mhd'. ('Mindesthaltbarkeitsdatum'). 15 abbreviations do not have an equivalent full form in the repository.

7.2.2.2 Adverbs Repartition

The figure 7.4 shows the repartition of distinct adverbs (blue) and the total of adverbs (red) over the business repositories. Three quarters of the adverbs (315) occur in only one database, whereas only 48 adverbs occur in at least two different databases. No adverb occurs in each database and only 1 occurs in 13 different databases: 'vorhanden'. This adverb represents 293 adverb occurrences, the second highest value. The highest value concerns the 315 adverbs which occur in only one database. They represent a total of 668 adverb occurrences, which means that, in average, each exotic adverb occurs only twice.

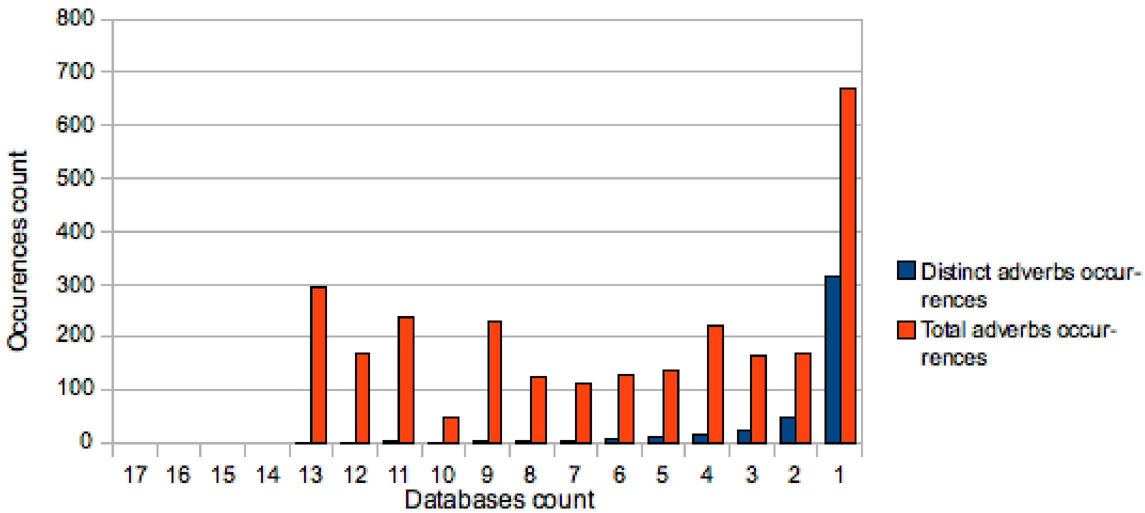


Figure 7.4: Repartition of adverbs over databases

The figure 7.5 shows the coverage of adverbs over the business repositories if they were registered in the knowledge base. If the most common adverb 'vorhanden' were registered (in green), nearby 11% of the adverbs would be covered (red line). With 6 additional verbs, about 26% would be covered. To reach a coverage of 70%, only 72 adverbs need to be registered, 120 to reach 75%. It would exclude the adverbs which occur in only one database, which represents an average effort of 18,5 adverbs per database.

A point of interest is to check whether the adverbs which occur only a few times are domain-specific. While examining manually these adverbs, it appears that they can be divided into 8 main different groups:

- Regular common adverbs: 'gering', 'inaktiv'

- Unusual synonyms of common adverbs: 'additional' instead of 'zusätzlich'
- Compound adverbs deriving from standard adverbs like 'technisch', 'bereit', 'fähig', 'spezifisch': 'abholbereit', 'produktspezifisch', 'sicherheitstechnisch'
- Compound adverbs deriving from grammatical suffixes like 'weise', 'bar', 'los': 'quartalsweise', 'machbar', 'erfolglos'
- Enumerated values like colors: 'blue', 'red'
- Abbreviations of common adverbs: 'urspr.', 'amtl.'
- Anglicisms: 'live'
- Very few domain-specific adverbs: 'chemisch'

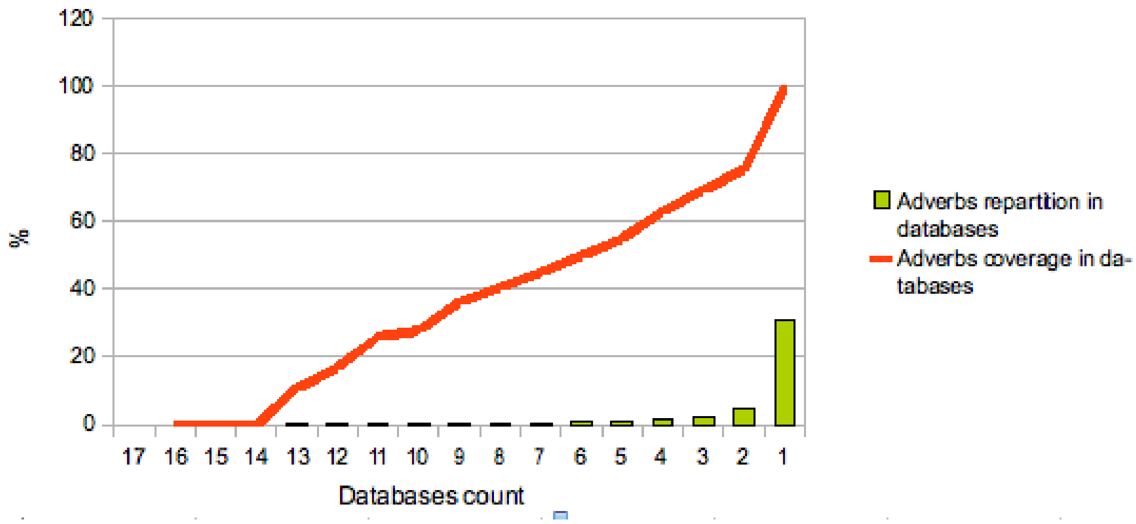


Figure 7.5: Coverage of adverbs over databases

This categorization of course counts for more frequent adverbs. In conclusion, one can say that only a few adverbs are domain-specific and the major parts must be registered in the standard knowledge base.

7.2.3 Modifiers Analysis

7.2.3.1 Global Statistic Results

When examining the structure of business expressions, activities or business terms are often associated to an adjective to refine their semantics like 'neuer Auftrag'. These adjectives are further called modifiers in the business object context. The first step of the modifiers analysis consists in extracting all adjectives which occur in events and functions from the corpus business repositories. The analysis is based on the results of the LE so that only the adjectives reduced to their base form are taken into account.

2734 adjectives were extracted, 593 are distinct ones. Only 2 adjectives have more than 100 hits. 54 adjectives occur between 10 and 99 times. 178 adjectives occur between 3 and 9 times. 135 adjectives occur only twice and 224 adjectives have only one single occurrence in the corpus. The results comprise also many redundancies with the statuses results like 'vorhanden': depending on the context, an adjective can be used as an adverb.

The table 7.4 shows the adjectives which occur the most frequently (20 first hits):

Adjectives	Total amount	Comment
neu	177	Opposite of 'alt'
weiter	123	Synonym of 'ander', 'sonstig'
intern	63	Opposite of 'extern'
technisch	58	
extern	51	Opposite of 'intern'
automatisch	39	Opposite form of 'manuell'
ander	35	Synonym of 'weiter', 'sonstig'
fehlend	35	
offen	35	
allgemein	32	
sonstig	28	Synonym of 'ander', 'weiter'
alt	28	Opposite of 'neu'
vorhanden	27	
geplant	26	Verb form
nahe	25	
manuell	24	Opposite of 'automatisch'
pers.	24	Abbreviated form of 'persönlich
fachlich	23	
schriftlich	21	
potenziell	21	

Table 7.4: Adjectives statistics

An interesting point is that, just as with statuses, the 20 first occurrences comprise many related terms: synonyms and opposite forms. When aggregating the synonyms values, 'weiter' reaches the first rank with 186 occurrences. The abbreviation 'pers' has 21 full form hits in the corpus. When aggregating the abbreviated form of 'pers.' the occurrence count increases to 45. If so much related terms are available in the 20 first hits, it is highly probable that similar cases occur in the remaining results. The first 20 hits also contain a lot of very unspecific terms: while 'neu', 'automatisch' are clear statuses, 'weiter', 'potenziell', 'ander', 'sonstig', 'allgemein' do not have a strong semantics. One of the modifier, 'geplant' directly derives from the verb 'planen'.

7.2.3.2 Adjectives Repartition

The figure 7.6 shows the repartition of distinct adjectives (blue) and the total of adjectives (red) over the business repositories. About three quarters of the adjectives

(415) occur in only one database, whereas only 65 adjectives occur in at least two different databases. No adjective occurs in each database and only 1 occurs in 12 different databases: 'neu'. This adjective represents 177 adjective occurrences. The 20 adjectives with the highest occurrences represent more total occurrences (911) than the 415 adverbs which are occurring in only one database. The latter represent a total of 845 adjectives occurrences, which means that in average each exotic adjective occurs only twice.

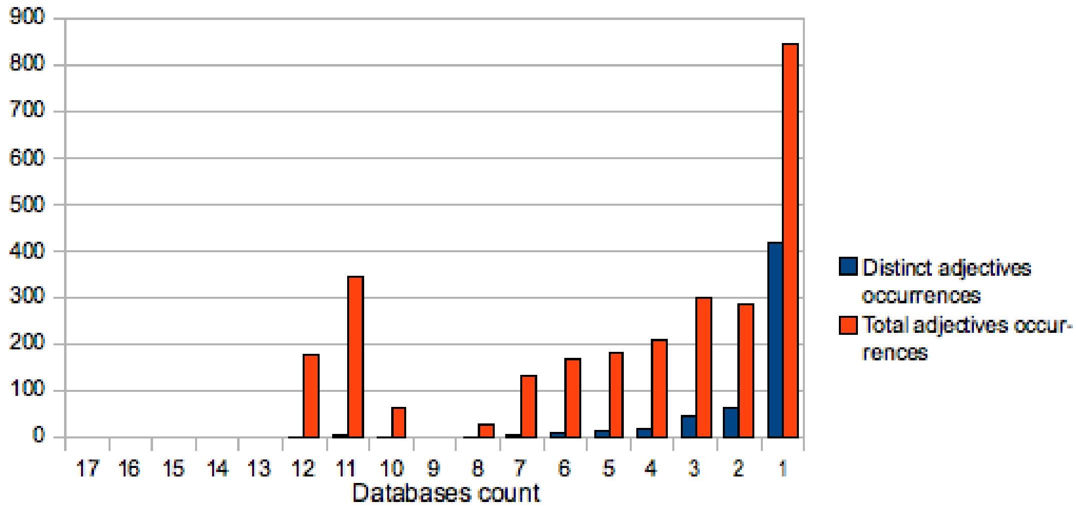


Figure 7.6: Repartition of adjectives over databases

The figure 7.7 shows the coverage of adjectives over the business repositories if they were registered in the knowledge base. If the most common adjective 'neu' were registered (in green), 6,5% of the adjectives would be covered (red line). With 5 additional adjectives, about 22% would be covered. To reach a coverage of 70%, 162 adjectives need to be registered. It would exclude the adjectives which occur in only one database, which represents an average effort of 24 adjectives per database.

Like for activities and statuses, the point of interest is to check whether the adjectives which occur only a few times are domain-specific. While examining manually these adjectives, it appears that they can be divided into 8 main different groups:

- Regular common adjectives: 'inkrementell', 'essentiell'
- Unusual synonyms of common adjectives: 'eventuell' instead of 'potenziell'
- Compound adjectives deriving from standard adjectives like 'bereit', 'fähig', 'spezifisch', 'relevant' (104): 'zeitkritisch', 'datenschutzrelevant', 'genehmigungspflichtig', 'sicherheitstechnisch'
- Compound adverbs deriving from grammatical suffixes like 'weise', 'bar', 'los': 'wochenweise', 'konfigurierbar', 'kostenlos'

- Enumerated values like colors: 'blue', 'red'
- Abbreviations of common adjectives: 'fach.', 'techn.', 'polit.'
- Anglicisms: 'customized'
- Some few domain-specific adverbs: 'gemeindeeigen', 'aufenthaltsrechtlich'

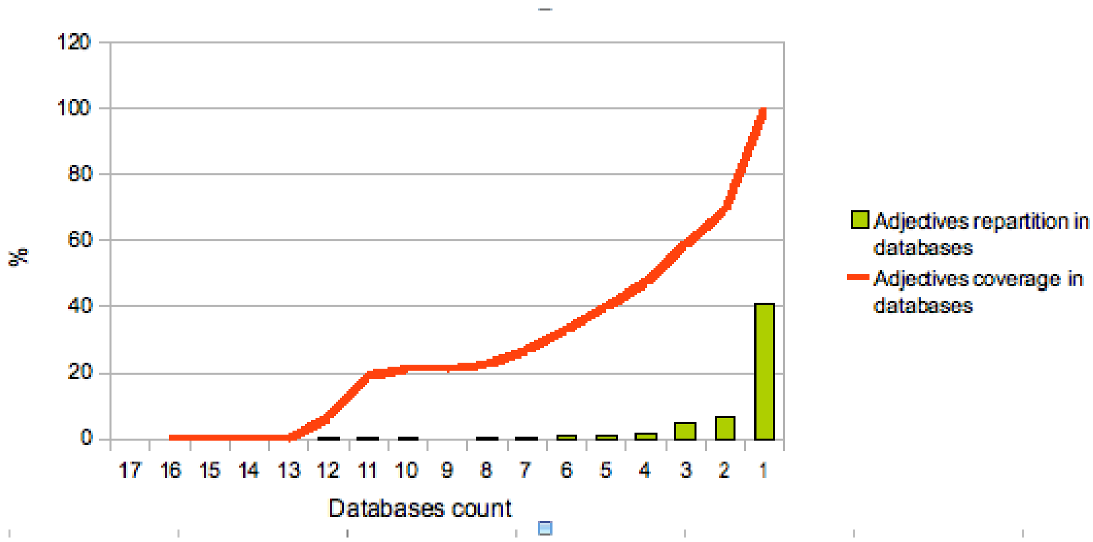


Figure 7.7: Coverage of adjectives over databases

This categorization of course counts for more frequent adjectives. The results are very analog to the adverb ones. In conclusion, one can say that only a few adjectives are domain-specific and the major parts must be registered in the standard knowledge base.

7.2.4 Nouns Analysis

The previous sections demonstrate that verbs, adverbs and adjectives mainly belong to a general business vocabulary and are only domain-specific in exceptional cases. With the noun analysis, it is expected that much more terms are domain-specific because nouns feature a strong semantics.

7.2.4.1 Global Statistic Results for Simple Terms

The first step of the nouns analysis consists in extracting all nouns which occur in events and functions from the corpus business repositories. The analysis is based on the results of the LE, which is able to differentiate the German compound terms from simple terms and to tokenize the compounds in their parts. The terms are reduced to their base forms, so that flections like plural or genitive forms are ignored by the statistic process. Due to the amount of nouns, the statistic process is first performed on simple terms and afterwards on compounds to obtain finer results. 22367 nouns (simple terms) were extracted, 2372 are distinct ones. One noun has more than 1000 hits. 22 nouns have more than 100 hits. 480 nouns occur between 10 and 99 times. 777 nouns occur between 3 and 9 times. 400 nouns occur only twice and 693 nouns have only one single occurrence in the corpus.

The table 7.5 shows the simple terms which occur the most frequently (20 first hits):

Simple terms	**Total amount**	**Comment**
Kunde	1053	Agent
Ag	303	Abbreviation, 'Aktiengesellschaft'
Job	302	
Auftrag	301	Unspecific term
Daten	290	Unspecific term
Rechnung	222	
Meldung	210	
Antrag	208	Unspecific term
Teil	192	Unspecific term
Angebot	174	Unspecific term
Status	161	
änderung	153	Action, derived from activity ändern
Fahrzeug	152	
Prüfung	141	Action, derived from activity prüfen
Unterlage	135	
Lieferant	132	Agent
Anforderung	117	
Kosten	116	
Bestellung	115	
Anfrage	113	

Table 7.5: Simple terms statistics

In contrast to verbs, adjectives and adverbs, no terms within the 20 first hits are related. The synonym 'Abnehmer' for the most used term 'Kunde' has 55 occurrences and is the 62th most used term. An interesting point is that the most frequent terms are very unspecific. It is not clear which kind of 'Daten', 'Antrag', or 'Angebot' is meant. The high amount of unspecific terms gives an idea of the lack of precision of the business objects denomination. 'änderung' and 'Prüfung' are terms which define an action. 'Kunde' and 'Lieferant' are used in business object denominations alternatively as subjects and as objects. This type of terms are further called 'agents'.

7.2.4.2 Simple Terms Repartition

The figure 7.8 shows the repartition of distinct simple terms (blue) and the total of simple terms (red) over the business repositories. About two-third of the simple terms (1577 of 2372) occur in only one database. No simple term occurs in each database and only 1 occurs in 14 different databases: 'Auftrag'. This simple term represents 301 simple terms occurrences. The 68 simple terms with the highest occurrences represent as much total occurrences (6179) than the 1577 adverbs which occur in only one database. The latter represent a total of 6189 adjectives occurrences, which means that in average each exotic simple term occurs four times, much more than exotic activities, adverbs or adjectives.

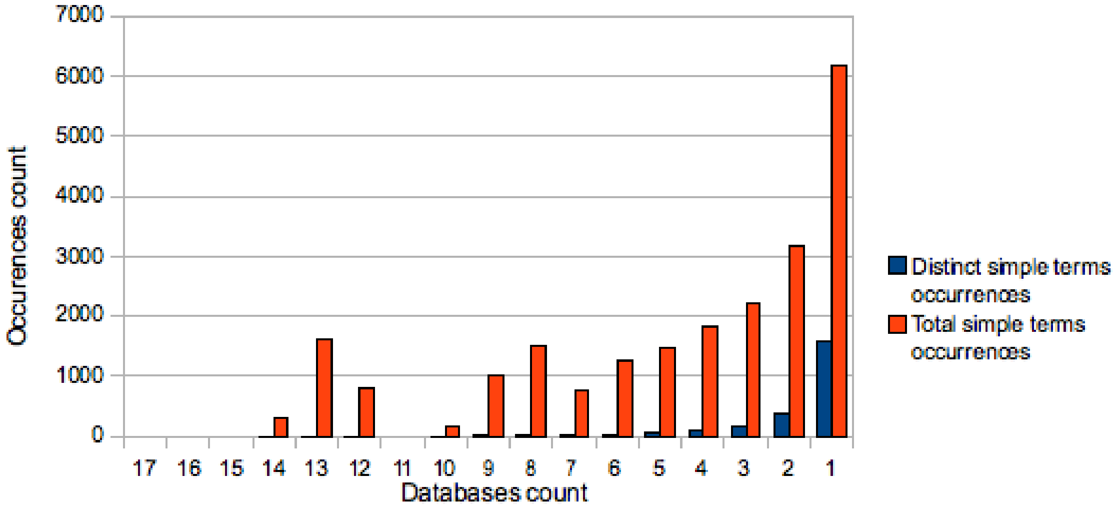


Figure 7.8: Repartition of simple terms over databases

The figure 7.9 shows the coverage of simple terms over the business repositories if they were registered in the knowledge base. If the most common simple term 'Auftrag' were registered (in green), only 1,35% of the simple terms would be covered (red line). This low value is explained by the fact that the term 'Auftrag', which occurs in the most databases, is not the term which has the highest number of total occurrences. With 4 additional simple terms, 8,62% would be covered. To reach a coverage of 72%, 795 simple terms need to be registered. It would exclude the simple terms which occur in only one database, which represents an average effort of 92 simple terms per database.

The point of interest is to check whether the simple terms which are occurring only a few times are domain-specific. While examining manually these simple terms, it appears that they can be divided into 14 main different groups:

- Regular common simple terms: 'Frist', 'Konzern', 'Parameter'
- Unusual synonyms of common terms: 'Check' instead of 'Prüfung'
- Time spans: 'Quartal', 'Woche'
- Agents: 'Abholer'
- Actions, ending with the suffix 'ung': 'Zusammenführung', 'Sperrung'

- Capabilities, ending with the suffix 'keit': 'Vollständigkeit', 'Machbarkeit', 'Umsetzbarkeit'
- Locations: 'Zentrale', 'Sekretariat'
- Cities (could also be countries): 'Saarbrücken', 'Berlin'
- Enumerated values like days: 'Montag'
- Anglicisms, not really domain-specific: 'firewall', 'scoring', 'provider'
- Abbreviations of common nouns: 'Art.', 'Std.', 'Transp.'
- Units, rarely domain-specific: 'Promille', 'Euro', 'p.m.'
- Acronyms, mostly domain-specific like 'SWP', 'UMG', sometimes more general: 'ERP', 'StVO'
- Some few domain-specific terms: 'Asylantrag', 'Strom'

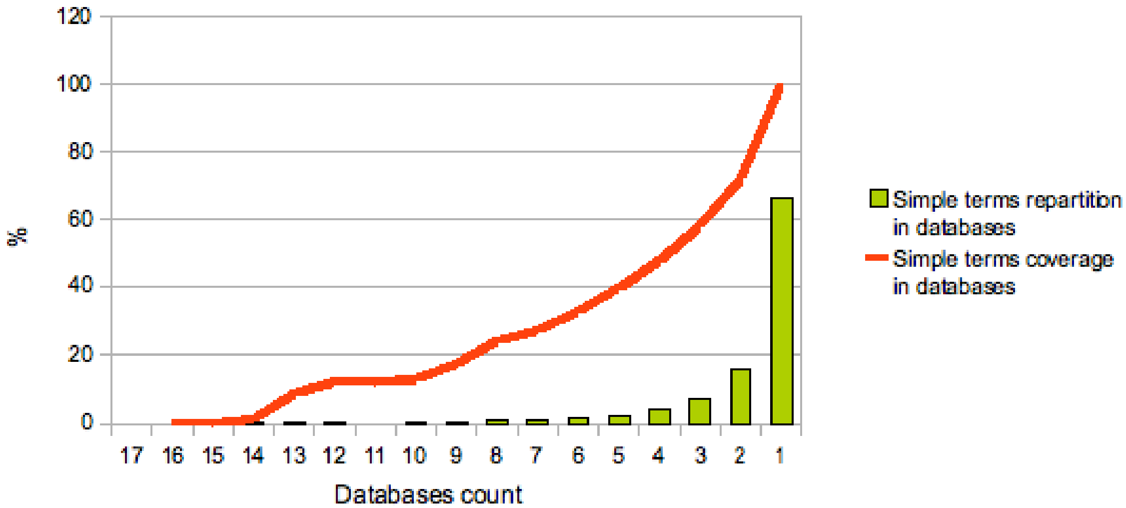


Figure 7.9: Coverage of simple terms over databases

This categorization of course counts for more frequent simple terms. Some of the categories give an additional semantics to the terms, like actions, agents, capabilities. In conclusion, one can say that most of the domain-specific simple terms are acronyms. Like seen in the occurrences statistics, a lot of simple terms are too unspecific to be really domain-specific. This explains why only some few terms are domain-specific. The major part of the terms, excluding acronyms, must be registered in the standard knowledge base.

7.2.4.3 Global Statistic Results for Compound Terms

The second step of the nouns statistics consists in extracting all compound terms which occur in events and functions from the corpus business repositories. 15776 nouns (compounds) were extracted (three-quarter of the amount of simple terms), 5354 are distinct ones (more than twice the amount of simple terms). Three compounds have more than 100 hits. 200 compounds occur between 10 and 99 times. 1414 compounds occur between 3 and 9 times. 1391 compounds occur only twice and 2346 compounds have only one single occurrence in the corpus.

The table 7.6 shows the compounds which occur the most frequently (20 first hits):

Compounds	**Total amount**	**Comment**
Transportauftrag	367	Logistic, derives from one of the most frequently used simple terms, 'Auftrag'
Fernverkehrstour	126	Logistic
Nahverkehrstour	116	Logistic
Packstück	72	Logistic
Ladeeinheit	71	Logistic
Anlagenabgang	68	Software and electro-technics industry
Kundenstammdaten	62	Specializes 'Stammdaten'
Wareneingang	61	Term used in the majority of databases
Gutschrift	52	Term used in 7 different databases
Störfall	50	Software company
Rahmenofferte	50	Logistic
Einzelofferte	46	Logistic
Stammdaten	45	Generic term for 'Kundenstammdaten'
Kundenanfrage	44	Term used in 7 different databases, contains two of the most frequently used simple terms
Arbeitsgang	44	Occurs in the 'banking' database
Werbemittel	43	
Lieferschein	36	Term used in 7 different databases
Datensatz	36	Term used in 4 different databases
Konditionsart	36	Term used in 2 different databases
Kundentransaktion	36	Distance selling

Table 7.6: Compounds statistics

In contrast to the activities, adverbs, adjectives and simple terms statistic results, the compounds which occur the most frequently are not automatically the terms shared by the highest number of databases. In the 20 first hits, 12 terms occur in only one database. The term 'Transportauftrag', which is a specialization of the most frequently used simple term 'Auftrag', and the term 'Kundenstammdaten', which specializes 'Stammdaten', which in turn specializes 'Daten', are of interest. Considering only the 20 first hits, the major part of compounds occurring in only one database is domain-specific, unlike the other word categories.

7.2.4.4 Compounds Repartition

The figure 7.10 shows the repartition of distinct compound terms (blue) and the total of compound terms (red) over the business repositories. This chart looks very different compared to the charts resulting from the activities, adverbs, adjectives and simple terms statistics. About 14/15 of the distinct compound terms (4767 of 5354) occur in only one database. No simple term occurs in each database and only 1 occurs in 9 from 17 different databases: 'Wareneingang'. This compound represents only 61 compound occurrences, corresponding to 0,39% of the total compound occurrences. The number of databases in which this term occurs is the lowest value from all statistics: 9. The 370 compounds occurring in at least two databases represent only a quarter of the total occurrences (3908 from 15776). The fact that three-quarters of the total occurrences (15776) occur in only one database means that in average each exotic compound occurs 3,3 times, close to the simple term value.

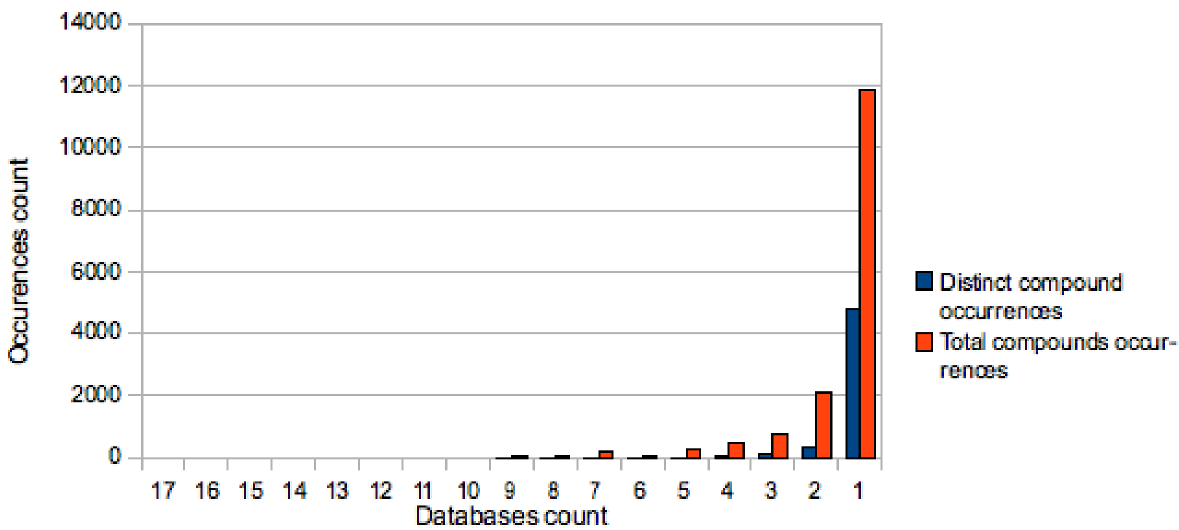


Figure 7.10: Repartition of compound terms over databases

The table 7.6 shows that a major part of the compounds occurring the most frequently are domain-specific. These terms occur in only one database unlike to the other statistics where the most frequently used terms were also the terms used in most of the databases. To refine the results, the table 7.7 shows the compounds which occur in the highest number of databases: the 30 first hits, appearing respectively in 9, 8, 7, 6 and 5 different databases. These terms represent the majority of the domain-unspecific compounds. An interesting point is that the following compounds are mainly composed of the most frequently used simple terms.

Compounds	Total amount	Databases count	Comment
Wareneingang	61	9	Derives from one of the most frequently used simple terms, 'Eingang'
Zahlungseingang	27	8	Derives from one of the most frequently used simple terms, 'Eingang'
Liefertermin	20	8	Composed of 'Termin', which occurs as a synonym ('Zeitpunkt') in this table
Kundenauftrag	72	7	Composed of two of the most frequently used simple terms, 'Kunde' and 'Auftrag'
Gutschrift	52	7	
Kundenanfrage	44	7	Composed of two of the most frequently used simple terms, 'Kunde' and 'Anfrage'
Lieferschein	36	7	
Rahmenvertrag	33	7	
Zeitpunkt	17	7	Often synonym of 'Termin', unspecific term
Neukunde	16	6	Composed of the most frequently used terms, 'Kunde' and one of the most frequently used adjectives, 'neu'
Kundenkontakt	15	6	Starts with the most frequently used simple term, 'Kunde'
Auftragsbestätigung	11	6	Starts with one of the most frequently used simple term, 'Auftrag'

Compounds	Total amount	Databases count	Comment
Zahlungsart	7	6	
Kundenangebot	13	5	Composed of two of the most frequently used simple terms, 'Kunde' and "Angebot'
Kundendaten	16	5	Composed of two of the most frequently used simple terms, 'Kunde' and 'Daten'
Schnittstelle	18	5	
Antragsteller	35	5	Agent
Stammdaten	45	5	Derives from one of the most frequently used simple terms, 'Daten'
Bestellmenge	19	5	
Bestelldaten	9	5	Derives from one of the most used simple term 'Daten'
Auftragsdaten	19	5	Composed of two of the most used simple terms 'Auftrag' and 'Daten'
Kreditorenrechnung	8	5	
Liefermenge	13	5	
Lohnsteuer	12	5	
Stichprobe	9	5	
Kostenrechnung	7	5	
Personalkosten	6	5	
Anfragebearbeitung	5	5	
Qualitätsprüfung	5	5	Action ('prüfen), mostly used with 'durchführen
Rechnungsprüfung	5	5	Action ('prüfen), mostly used with 'durchführen or alone

Table 7.7: Compounds occurring in the most databases

The figure 7.11 shows the coverage of compound terms over the business repositories if they were registered in the knowledge base. Also, the coverage chart looks very different from the other coverage charts. If the compound appearing in the most databases 'Wareneingang' were registered (in green), only 0,39% of the total compounds would covered (red line). This value which is the lowest one is explained by the fact that this term has not many occurrences (61) comparing to the 15776 total occurrences. With 12 additional compounds, only 2,32% would be covered. When registering all terms occurring in at least two databases (370), the coverage value reaches only 24,77%, far away from the minimum of 70% of all other categories and the 94% for the verbs. Registering the compounds which occur in only one database represents an average effort of 280 compound terms per database.

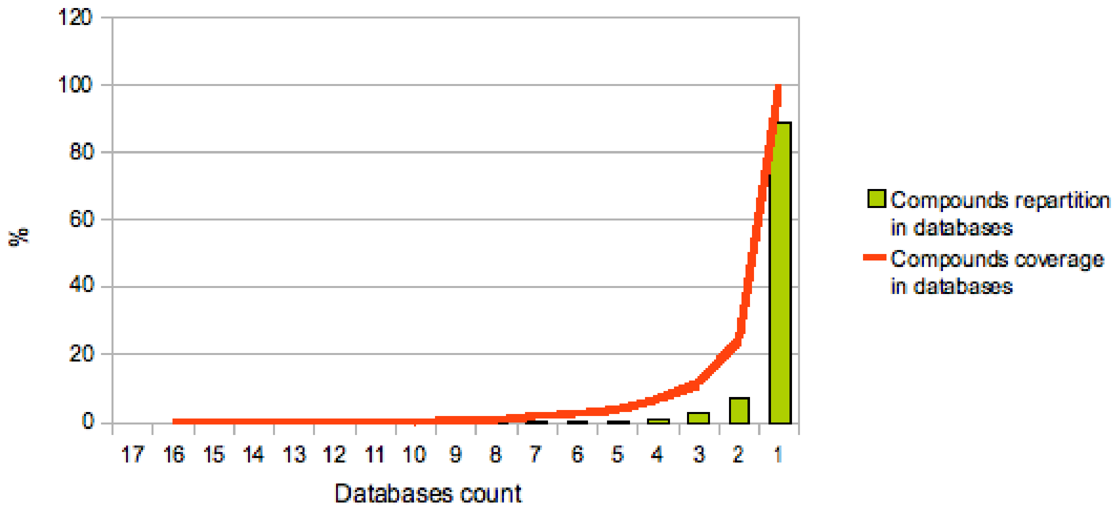


Figure 7.11: Coverage of compound terms over databases

The table 7.6 gives some answer elements to the question whether the compounds occurring in only one database are domain-specific. When examining manually the terms occurring in only one database, it appears that the semantic term classification is very similar to the simple terms one so that it will not be repeated here. Instead, the compounds structure as well as the domain classification are deeper examined.

The compounds structure can be divided in several categories:

- Compounds composed of very general simple terms, rarely domain-specific: 'Kreditunterlage', 'Anfrageformular', 'Kundenkontaktbearbeitung'
- Compounds composed of simple terms synonyms: 'Checkliste' instead of the more used 'Prüfliste', 'Qualitätscheck' instead of 'Qualitätsprüfung', 'Bruttolohn' vs. 'Bruttogehalt'
- More rarely, compounds composed of abbreviated forms of common simple terms: 'Anwendungsdoku' instead of 'Anwendungsdokumentation'
- Compounds starting with domain-specific acronyms: 'CRC'-Workshop'
- Domain-specific compounds like following examples:

- Chemical industry: 'Basisrezept' and its variant 'Grundrezept', 'Teilbasisrezept', 'Grundrezeptoperation', 'Analyseoperation'
- Automotive supplier: 'Endmontage', 'Werkzeugprobe', 'Serienmaschine'
- Communal administration: 'Parkplatz', 'Parkausweis', 'Fahrzeugdaten' and its variant 'Kfz-Daten', 'Wohnungsverwaltung', 'Sozialamt'
- Insurances: 'Schadenshöhe', 'Risikoanpassung', 'Kulanzhöhe'
- Logistic: 'Fernverkehrsfahrzeug', 'Fernverkehrstourvorlage', 'Gangsspediteur'

This list shows that compound terms are really the most domain-specific terms. The few examples point out many inconsistencies where some compound parts are synonyms: 'Lohn', 'Gehalt', 'Basis', Grund', 'Fahrzeug', 'Kfz'. Not considered are inconsistencies due to different hyphenations ('Grobplan' vs. 'Grob-Plan') or grammatical derivations ('Installationstermin' vs. 'Installierungstermin') because the base form reduction of the compound parts resolves these cases. Finally, domain-specific terms often have related terms in the same database, like the different terms based on 'Rezept', or 'Fernverkehr'.

Following method/application helps to classify compound terms between domain-specific terms and general business vocabulary:

1. Filtering out rare synonym terms of general terms like 'Checkliste'. If the knowledge base provides an exhaustive and accurate list of simple terms including synonym relations between them, the compound variant parts can be replaced by the synonym parts and synonym compounds are deduced.
2. Filtering out compounds composed of very general simple terms. Simple terms are considered to be very general if they are occurring in several databases from different domains. This means that these terms belong to a global business vocabulary set. To achieve this filtering, each simple term must be annotated with a weight, deduced from usage statistics like in the previous sections.
3. Collecting compound terms containing domain-specific acronyms.
4. If several databases from the same domain are available, collecting all terms occurring in these databases which are not part of databases from other domains (excluding the points 1 and 2).
5. Collecting terms which have a high occurrence count, but only in the considered database like the logistic terms in the table 7.6.
6. Dividing the remaining compounds in their parts and make an occurrence statistic of the parts. Collecting the compounds in which the most common parts are occurring. This should give a set of related terms. Performing this step in the logistic database results in the most frequently use compound parts 'Verkehr', 'Laden', 'nahe', Tour', 'senden', 'Offerte', 'fern', 'Transport', 'Lager', 'entladen'.

7. If a domain-specific knowledge base is already available, searching for terms related to the knowledge base ones:
   - More specific ones, checking additional compound parts: 'Nahverkehrstour' is more specific than 'Verkehrstour'
   - Related ones, where one compound part differ: 'Nahverkehrstour' vs. 'Fernverkehrstour'
8. The remaining compounds should be considered manually because they can either be a general business term which is not so commonly used or a domain-specific term which occurs only rarely in the database. This remaining list is provided automatically and the business content expert decides in which knowledge base the term should be registered.

7.2.5 Summary of the Statistic Key Values and Conclusion

The table 7.8 summarizes the previous statistic results, pointing out the huge effort it represents to register all terms with manual techniques only, knowing that the total of distinct term occurrences represents 9755 for all term categories for the 17 examined databases. With each additional database, it is expected that new terms occur. However, the more terms are registered, the more the average effort per database should decrease, in particular if they are issued from analog domains.

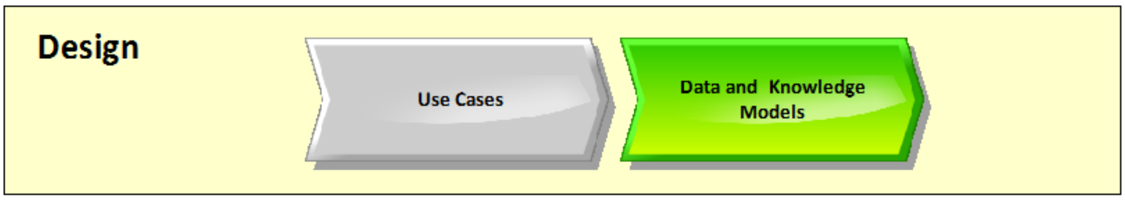
	Verbs	Adverbs	Adjectives	Simple terms	Compounds
Total occurrences	27403	2705	2734	22367	15776
Total distinct occurrences	1017	435	577	2372	5354
Average occurrence small per term	26,94	6,22	4,74	9,43	2,95
Maximum term occurrence	4189	293	177	1053	367
Effort if terms occurring in at least 2 databases are registered	30 terms	18 terms	24 terms	92 terms	280 terms
Term coverage if terms occurring in at least 2 databases are registered	94,19%	75,3%	69,09%	72,33%	24,77%

Table 7.8: Key values comparison

Each domain has common processes like human resources, travel expenses, customer support and thus share common vocabulary, which should be registered in a common knowledge base called the business management language workspace. The difficulty resides in the fact that each company uses a certain amount of domain-specific vocabulary: each company has a certain logistic concern, even if the logistic service is delivered by an external provider. Each company may model some legal procedure without being able to evaluate some legal specialties. A cut must be done which separates these kinds of terms called 'low-level' domain-specific terms and the specialized terminology called 'high-level' domain-specific terms. The low-level domain-specific terms are often used in databases issued from different domains, so that the previous statistic methods really help to do this cut. In the previous statistics, no difference was made between domain and industrial sector because of the lack of reference material. In a later stage, if more databases issued from the same domain and the same industrial sectors are available and if some reference knowledge bases are implemented, it would be interesting to study whether terms can be classified into domain-specific and industrial sector-specific knowledge bases.

In summarizing, it can be stated that the method developed to extract domain-specific compounds is based on the differential between all occurring terms and general business terms. If domain-specific knowledge bases are exhaustive enough, the same method can be applied to extract company specific terms. This would be also very useful for acronyms, which are difficult to classify between domain and company-specific. Compounds including the company trademarks for instance are company-specific. Like seen in the previous sections, the only real domain-specific term category is the category of compound terms. If adverbs or adjectives are not part of the business management language workspace, they are with the utmost probability company-specific instead of domain-specific. Color names are a good example: each car manufacturer defines its own color names like 'babylon red'. A printing machines manufacturer would define a broad color palette, which is also not part of the common business language.

7.3 Second Dimension: Relations between Terms and Concepts



Up to now, the objective is to develop the common business management language workspace. The summarized statistics show the breadth of the term pool to be registered. The difficulties to code manually this amount of terms are:

- The expensive time effort
- The inaptitude of the business content expert to remind all the terms already registered to detect all the potential synonyms.

- The need of sustainability: once the knowledge bases are in use, it is difficult to update older entries because modeled business objects may be out-of-date, which limits the interoperability even within the company. As a consequence, the entries need to be accurate and reliable.

The previous statistics pointed out some relations between terms and some semantic categorizations. The knowledge acquired from these statistics and the requirements issued from the different use cases descriptions will be further used and analyzed in more detail to develop:

- Helper and automation methods in order to build the knowledge bases.
- Datatypes to describe the knowledge base structure (data model) which is able to manage all the needed relations and classifications.

These methods, combined with the statistic method to extract domain-specific vocabulary, can also be applied to develop domain-specific or company-specific knowledge bases.

The knowledge base aims to fulfill the linked data principles from Tim Berners-Lee, further detailed in the section 3.3. In summary, the four linked data principles:

1. URIs will be used to identify things.
2. HTTP URIs will be used so that these things can be referred to and looked up by people and user agents.
3. Provide useful information about the thing when its URI is dereferenced, using standard formats such as RDF/XML.
4. Include links to other URIs to improve the discovery of other related information.

The chosen KB format is the Web Ontology Language (OWL, see section 3.3.2 for more details about OWL), to benefit from the semantic advantages and features of ontologies and from a good software support: the KB will be created and administrated with Protégé (see 4.4.2). OWL fulfills the three first linked data principles: a unique name must be given for each OWL resource, resources can be addressed over the web via http and OWL itself is based on RDF/XML. This will simplify the implementation of the community business content collaboration platform. The defined relations between the terms allow the fulfillment of the fourth principle.

7.3.1 Knowledge Base Structure and Datatypes

7.3.1.1 Categories

The statistics and the use cases pointed out that terms are classified in different categories:

- Nouns, further called objects in the KB context. Independently of simple or compound terms, they can be divided in following categories:

  - Objects
  - Agents
  - Capabilities
  - Actions
  - Locations
  - Cities
  - Units
  - Trademarks
  - Time spans
  - IT systems
  - Organization units

- Activities, further called functions in the KB context. Denoted in statistics by verbs reduced to their base form, they are used in the infinitive form in business object denominations.

- Statuses, denoted by verbs, adverbs or adjectives. Adjectives act as a modifier for verbs or nouns and they are mostly used in their inflected form. The major part of the adjectives is redundant with adverbs and in fact, if a noun or a verb is modified by an adjective, the adjective acts also as a status for the modified object. For these two reasons, adverbs and adjectives will be treated together in the KB, as a sub-category of statuses. The difference between modifiers and statuses will be made later when outlining their usage. The statuses can be divided in following categories:

  - Verbs used in the past participle form in events, mainly resulting from a process step
  - Statuses having an opposite status: 'vollständig', 'unvollständig'
  - Enumerated values, where colors are the only identified enumeration in the statistic results
  - All other adverbs or adjectives like 'quartalweise', 'technisch'

- From the use cases, a comparator category is also needed which consists of mathematical comparators ('<', '='...) in combination with natural language terms (kleiner, gleich...).

- To allow an aspect-oriented navigation over terms, a process map structure must be available.

These categories, even if not so exhaustive as standard ontologies concepts, fulfill the needs to put a semantic layer on business object denominations. Acronyms and abbreviations are dependent of their long form and do not belong to a special category. Anglicisms are mostly used as a synonym of German terms in the analyzed context, so that they do not constitute a special category.

7.3.1.2 Attributes

Like the categories, use cases and statistics pointed out the need for associating some attributes to business terms:

- Language: although this dissertation concerns only the business object denominations in German language, the data structure must fulfill prerequisites to extend this concept to other languages. As a consequence, each term must be associated to its language.

- Weight, represented by an integer range from 0 to 100: the weight is needed in the controlled modeling use case (see 6.2.1.2) in order to build the cloud displaying the most used terms and in the algorithm helping to separate the general terms from the domain-specific ones. The initial weight value is issued from the previous statistics. When a term occurs in many databases and has a high occurrence value, the weight will trend to 100. Domain-specific KBs have their own weighting: if a term occurs in a certain domain database, but has a high occurrence value, the weight will also trend to 100. If the business content expert activates several libraries, the cloud will consist of a mix between general terms and domain-specific terms. New terms must be weighted dependently of their occurrence count and usage domain.

- Usage form: like seen in the use cases controlled or restricted modeling, the system must propose the most appropriate term when several terms are synonyms. The values to represent the usage form are standard values used for terminology maintenance: 'preferred form', 'accepted form' and 'deprecated form'. As far as possible, the accepted form must be avoided. This value is mainly used to allow units abbreviations.

- Abbreviations: a synonym of a term can be abbreviated, particularly for units. A flag is needed to specify whether a term is an abbreviated form. Combined with the usage form attribute, one can declare whether the abbreviated form is allowed.

- Technical terms: like seen in the SOA use case, terms must have an equivalent technical term to allow the transformation of a business process into an IT process which respects naming conventions for technical interfaces. In some exceptional cases, it is wished that the terms are used in the plural form. This must be avoided as far as possible because in the majority of cases, plural forms can be represented by loops in the process model design. Anyway, to treat this possibility, the technical term must have an equivalent technical plural term.

- Linguistic features: the LE contributes to the different business object denomination checks and operations. Some features of the LE, like the base forms of terms, the compounds parts for a compound term and a citation form must be associated to each term. While this process can be automated, datatypes must be provided to save the corresponding values.

7.3.1.3 Relations

The use cases and the statistics emphasized several relation types between terms. Two kinds of relations are distinguished: static relations and dynamic relations.

Static relations describe sustainable facts about terms, like semantic or grammatical relations. Following static relations are identified:

- Synonyms ('senden', 'versenden', 'schicken'...), including abbreviations that are considered as synonyms of full forms.
- Negated forms ('möglich', 'unmöglich')
- Relation between functions and verb statuses ('prüfen', 'geprüft').
- Relation between functions and nouns ('Ende', 'beenden')
- Relation between functions and actions ('anmelden', 'Anmeldung')
- Relation between statuses and capabilities ('machbar', 'Machbarkeit')
- Specialization of terms: 'Auftrag' is more specific than 'Kundenauftrag'.

As a chosen design pattern, synonyms will not be defined as relations between terms, but they will be grouped into concepts called business concepts instead. Concepts are no concrete natural language instances. They are issued from a reification process: natural language terms are mapped to a concept in order to create an identifier in the KB, which can be further processed. This makes the concept independent of the language and allows synonyms to be declared as items (individuals in OWL) of a given concept. The concept then acts like a class in object-oriented programming languages. A class groups objects having same characteristics, in this case the same or a very close semantics. The advantage of this representation is that other relations on concept level can benefit from this information. For example, defining that the concept of 'prüfen' is related to the concept of 'geprüft' and defining that 'prüfen', 'checken', 'überprüfen' are synonyms avoid declaring again the statuses 'geprüft', 'gecheckt', 'überprüft' as synonyms.

Dynamic relations describe the context usage of terms:

- Which functions can be applied or never applied to which objects.
- Which statuses can apply or never apply to objects.
- Which modifiers apply or never apply to an object or a function.
- A link from terms to process map elements is necessary, to provide an aspect-oriented navigation for business terms.
- Relations between process map elements to define the superior and inferior elements.

All these relations can be extended if required. The only restriction is that the data must be always downwards compatible. Additional relations, attributes or categories are not problematical, but changes to the existing structure must be avoided.

7.3.1.4 Resulting Structure and Datatypes

The identified categories, relations and attributes result in classes and datatypes in an ontology built with OWL. Four types of elements can be distinguished in this ontology:

1. The built-in root class of all OWL ontology classes 'owl:Thing'.
2. The container classes for business concepts which act as datatypes.
3. The business concepts, which are subclasses of datatype classes.
4. The individual elements which are leaf nodes associated to business concept classes and which are the natural language occurrences in business object denominations.

The UML class diagram 7.12 shows the basic datatypes, designed as a container for business concepts and the needed static relations. The overlapping generalization relations are highlighted in blue, the overlapping '*hasCorrespondingFunction*' in red.

The class '*AbstractBusinessConcept*' is the root class describing business concepts and is a direct subclass of 'owl:Thing'. This class is named as an abstract class. It means that no concepts should be directly associated to this class. However, no direct mechanism is provided in OWL to define a class as abstract. Actually, it does not really make sense in OWL because the notion of what class something belongs to is a lot more fluid. This can change with additional assertions or inference, so there is not the same degree of concern as with object-oriented programming language. It is why classes which act as an abstract datatype in this concept are as convention prefixed with 'Abstract'.

The class '*AbstractBusinessConcept*' contains attributes that are valid for all type of business concepts or for all individuals:

- *language*: identified by the natural language code as defined by Alvestrand [99]. It applies to all, and only to, individuals because they are the only language-specific items in this ontology.
- *usageForm*: of type *'UsageForm'*, an enumeration which comprises the three values 'PreferredForm', 'AcceptedForm' and 'DeprecatedForm'. It applies to all, and only to, individuals. An individual marked as 'PreferredForm' is the favored term for a given concept. 'AcceptedForm' means that in some contexts this language-specific instance is right. 'DeprecatedForm' should always fire a warning to the user in order to reformulate the business expression with the preferred form or potentially with the accepted form.

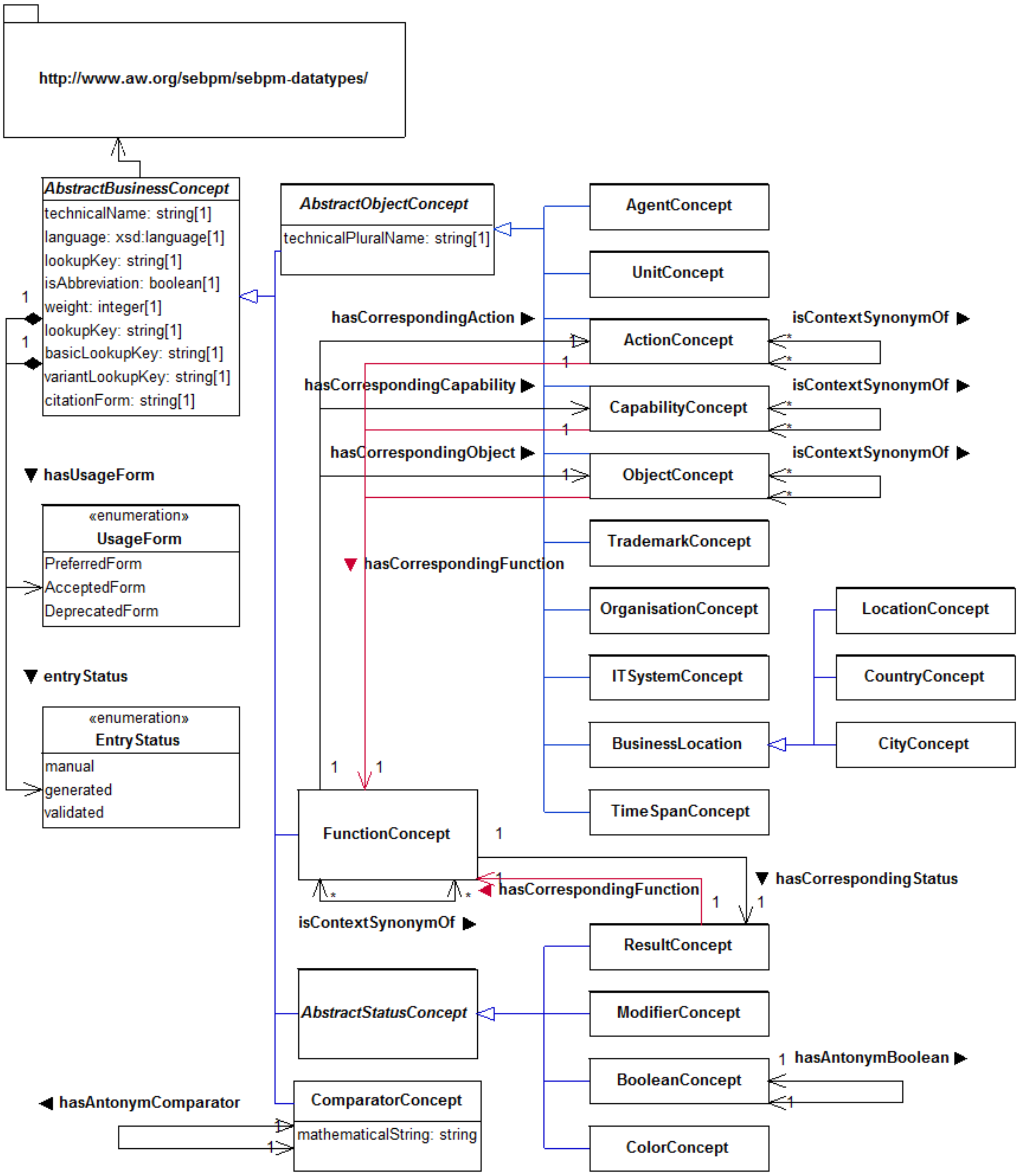


Figure 7.12: Datatypes for SE-BPM concepts

- *isAbbreviation*: this Boolean attribute applies to all, and only to, individuals. A value set to 'true' indicates that the given individual is an abbreviated term. If no value is set, it is considered to be 'false'.
- *technicalName*: string which applies to all, and only to, concepts because the technical name is unique for one concept and independent of the number of individuals associated to the given concept. Detailed explanations for the technical term usage are given in the section 8.3.6.

- *weigh*: integer, defining the occurrence probability of the term.

- *citationForm*: string provided per default by the LE, but can be manually adapted if needed. Denotes the preferred writing form of a term. It is specific to terms, not to concepts, so that there is no need to adapt the preferred form on concept level if a term status is changed from deprecated to preferred.

- *lookupKey*: normalized string provided by the LE, used to match the terms used in business object denominations with the ontology concepts. It applies to all, and only to, individuals. This attribute value will be set automatically after a LE analysis in order to ensure the correct matching between this value and the LE analysis result. The lookup key is reduced to the base form, has separated compound parts and the hyphen positions are deleted.

- *basicLookupKey*: normalized string provided by the LE. This key is not used for matching purposes, but it gives more grammatical information about a term like the base verb form for a noun or whether a verb has a separable prefix. It applies to all, and only to, individuals. This attribute value will be set automatically after a LE analysis process.

- *variantLookupKey*: normalized string provided by the LE. This key is not used for matching purposes, but to detect writing variants of terms. This string is reduced to the base form, has separated compound parts, but keeps hyphenation positions and keeps some derivation information. This attribute value will be set automatically after a LE analysis process.

- *entryStatus*: of type *'EntryStatus'*, an enumeration which comprises the three values 'manual', 'generated' and 'validated'. It describes how the individual was created. If the business expert has created it manually, 'manual' is set. A validation step can be optionally performed after a manual step, for instance, if the creator was not an expert. If the system created automatically the entry, the value is set to 'generated'. A validation step is preferred in this case.

The abstract class '*AbstractObjectConcept*', which inherits from '*AbstractBusinessConcept*', is a container for all classes and corresponding concepts that can stand for a business object. These concepts are mainly nouns (simple terms, compounds) and noun phrases (NP). A plural technical name is required for this category, defined by a string attribute named '*technicalPluralName*'. Like the attribute '*technicalName*', it concerns all and only concepts. This class contains several subclasses, which are concrete classes allowing concept members. The only difference between these subclasses is a concern of semantics. They group the different categories identified above. Underlying concepts and individuals share the same characteristics and attributes.

- *ObjectConcept*: container collecting standard objects, which do not belong to any other category.

- *AgentConcept*: container collecting agents.

- *ActionConcept*: container collecting actions.
- *CapabilityConcept*: container collecting capabilities.
- *UnitConcept*: container collecting units.
- *TrademarkConcept*: container collecting trademarks.
- *OrganisationConcept*: container collecting organization units.
- *ITSystemConcept*: container collecting IT system names.
- *BusinessLocation*: container class for different types of locations, in order to separate them semantically, but keep a possibility to handle all location types in an identical manner.
  - *LocationConcept*: all locations excepting cities and countries.
  - *CountryConcept*: container class collecting countries
  - *CityConcept*: container class collecting cities
- *TimeSpanConcept*: container class containing time spans. Day or month names will not be treated here because they do not occur in the corpus. They would be stored in additional classes.

The concrete class '*FunctionConcept*' inherits from '*AbstractBusinessConcept*' and is a container for all concepts representing a business activity (verbs).

The abstract class '*AbstractStatusConcept*', which inherits from '*AbstractBusiness-Concept*', is a container for all classes and corresponding concepts which can stand for a status or a modifier. This class contains several subclasses, which are concrete classes allowing concept members:

- *ResultConcept*: this concrete class acts as a container for all statuses concepts which are the result of an activity, mainly verb statuses.
- *BooleanConcept*: this concrete class is a container for all concepts representing a status or a modifier which have a clear opposite concept. For instance, the concept 'REQUIRED' has an opposite concept 'OPTIONAL'. This oppositional relation must be bi-directional.
- *ColorConcept*: this class is a container for all standard colors required in the standard KB. Can be extended with specific colors in customer-specific KBs.
- *ModifierConcept*: this concrete class groups all other concepts representing a status or a modifier which are no verbs, no colors and have no clear opposite concept. For example, one can not say that the opposite of 'poor' is 'rich' because a lot of nuances exist between 'poor' and 'rich'.

The concrete class '*ComparatorConcept*' inherits from '*AbstractBusinessConcept*' and is a container for all concepts representing a comparator which has a clear opposite comparator, like the mathematical comparators '$\leq$' and '$>$'. The opposite comparator is defined by a relation. This class contains an attribute, '*mathematicalString*', which applies to underlying concepts. The mathematical string cannot be an individual because on one hand it is language independent so that it makes no sense to set a language value and on the other hand special characters are not allowed as an OWL URI.

The table 7.9 defines the static relations needed between the classes. The sign '<–>' denotes a symmetric relation.

Source class	**Relation name**	**Target class**
FunctionConcept ObjectConcept ActionConcept CapabilityConcept	isContextSynonymOf	FunctionConcept ObjectConcept ActionConcept CapabilityConcept
BooleanConcept	hasAntonymBoolean <–>	BooleanConcept
ComparatorConcept	hasAntonymComparator <–>	ComparatorConcept
FunctionConcept	hasCorrespondingStatus	ResultConcept
FunctionConcept	hasCorrespondingCapability	CapabilityConcept
FunctionConcept	hasCorrespondingAction	ActionConcept
FunctionConcept	hasCorrespondingObject	ObjectConcept
ResultConcept ActionConcept CapabilityConcept ObjectConcept	hasCorrespondingFunction	FunctionConcept FunctionConcept FunctionConcept FunctionConcept
ActionConcept AgentConcept CapabilityConcept ObjectConcept	specifies	ActionConcept AgentConcept CapabilityConcept ObjectConcept
ActionConcept AgentConcept CapabilityConcept ObjectConcept	generalizes	ActionConcept AgentConcept CapabilityConcept ObjectConcept

Table 7.9: Summary of static relations between classes

The table 7.10 defines the dynamic relations needed between the classes. Process map classes will be completed and explained in more detail in the section 7.4. The dynamic relations are created and extended in a collaborative way supported by usage statistics like described in the use cases. The relations involving the concepts of type '*AbstractObjectConcept*' are inherited by more specific terms like displayed in the figure 6.45. To avoid the inheritance, the relations starting by 'never' must be employed. The latter are also used to describe relations that are once validated to be negative by the business content expert to prevent from further analog relation requests by the business modelers.

Source class	**Relation name**	**Target class**
AbstractObjectConcept	associatedTo	FunctionConcept
AbstractObjectConcept	neverAssociatedTo	FunctionConcept
AbstractObjectConcept	associatedTo	AbstractStatusConcept
AbstractObjectConcept	neverAssociatedTo	AbstractStatusConcept
AbstractObjectConcept	modifiedBy	AbstractStatusConcept
AbstractObjectConcept	neverModifiedBy	AbstractStatusConcept
FunctionConcept	modifiedBy	AbstractStatusConcept
FunctionConcept	neverModifiedBy	AbstractStatusConcept
AbstractObjectConcept	belongsToProcessMap	ProcessMap
ProcessMap	containsObject	AbstractObjectConcept

Table 7.10: Summary of dynamic relations between classes

All these datatypes are defined in the namespace URI 'http://www.aw.org/sebpm/sebpm-datatypes' to ensure uniqueness of the datatype names and for more maintenance flexibility. All concepts are defined in the namespace URI 'http://www.aw.org/sebpm/sebpm-concepts' as an additional OWL project, which imports the datatypes. This design decision has the advantage to store all language independent items in a separate file.

Concept names could be abstract names, independent of any natural languages, like numbers or a random alpha-numeric string. However, the person who has to work with the concepts cannot memorize so many abstract concept names and cannot imagine which terms are associated to this concept. It is why some conventions are defined for the concept naming through this dissertation:

- For clearness, the concept names are written in the English language, which is also the main language used for programming. They must be as close as possible to the technical name property.
- The concept names are always written in capital letters, to distinguish them from datatypes and individuals.
- Verbs start with the prefix 'TO_', in order to distinguish the verbs from nouns. For example, the noun 'start' and the verb 'to start' would have the same concept name 'START'. Protégé cares for uniqueness in class and individual names (even if OWL allows duplicate names).
- If a real and motivated need for duplicate names occurs, the concept names are suffixed by '_n', with 'n' being an integer to ensure uniqueness.

The UML class diagram 7.13 shows examples of concepts belonging to different datatypes. The datatype is displayed as a stereotype of the concept classes. For example, the concept 'ACCOUNT' has a stereotype 'ObjectConcept' and two attributes 'technicalName' and 'technicalPluralName' which have respectively the values 'account' and 'accounts'.

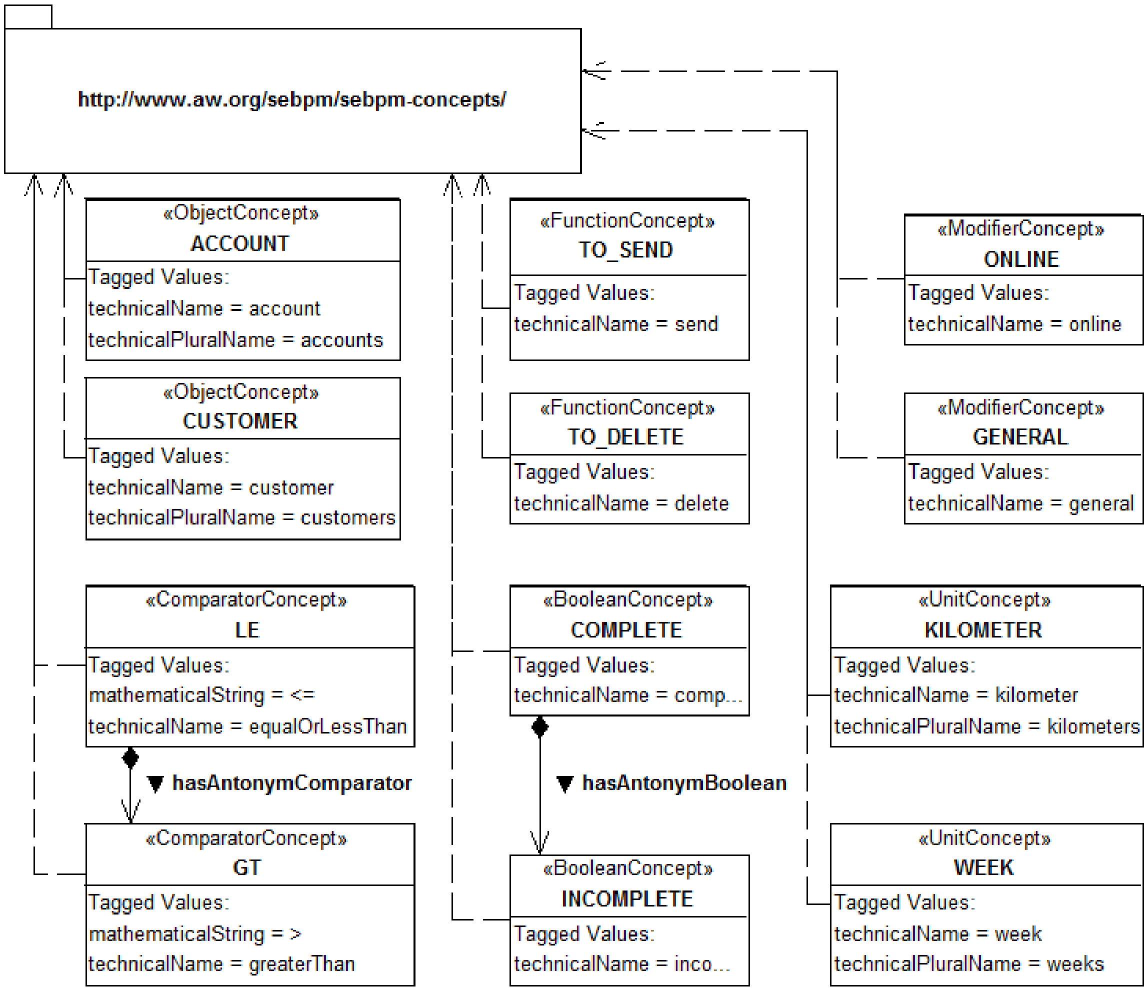


Figure 7.13: Concept examples

Individuals are unique language-dependent instances of the concept. They are defined in an other OWL project, which imports the concept ontology. This has the advantage of maintaining different language instances of the concepts in different files, to avoid the mixing of individuals issued from different languages because it increases the unique naming problems. For example, the individual 'nuance' exists in German, French and English and a naming strategy must then be defined to ensure uniqueness, like to prefix the language code. If the different languages are maintained in different files, they can be loaded at runtime in different language specific models and the language clash is avoided. This design decision increases also the visibility of individuals because their names can be kept as simple as possible. Like concepts, the individual names could be abstract names, but to fulfill the visibility requirement, following naming conventions are applied:

- In general, individuals are written in small letters and correspond as far as possible to the base citation form of the term.
- They are starting with a capital letter if they are proper names or German nouns.
- In case of trademarks or acronyms, the individual names remain as close as possible to the citation form .
- Due to the Protégé uniqueness requirement, if more individuals share the same name (for example 'warten' and 'warten' which are associated to the concepts 'MAINTAIN' and 'WAIT'), they are as a convention suffixed by '_n', with 'n' being an integer.

It is worth to note that individual names cannot stand for the lookup key because lookup keys can be redundant. For the same reason, they cannot stand for the citation form. Individual names are then only considered as a key in the KB associated to properties that store all the information necessary to further processing.

The UML class diagram 7.14 shows some examples of individuals. The examples in the first column are four individuals associated to the context 'SEND'. The term 'senden' is defined as the preferred form and 'verschicken', 'schicken' and 'versenden' as the deprecated forms. In the last column, the concept 'KILOMETER' has two instances, 'km' and 'kilometer'. The instance 'km' is declared as an abbreviation and as deprecated.

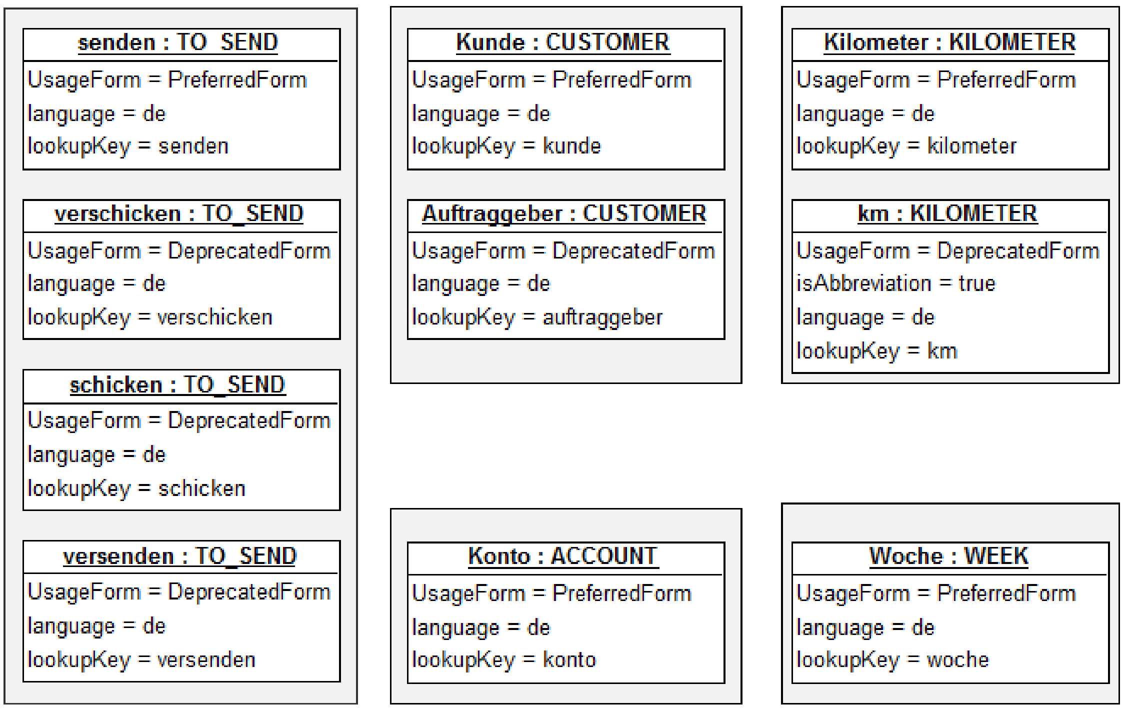


Figure 7.14: Individual examples

7.3.2 Supporting Techniques for Feeding the Knowledge Base

7.3.2.1 Supporting Techniques for Registering Activities

Verbs are the most domain-unspecific category. As a consequence, all verb occurrences, except from modal verbs which do not feature a semantics, must be registered in the standard KB. At this point, no entry is available in the KB, so that the KB feeding cannot base on pre-existing knowledge. This step is bound to the maximum manual work. However, supporting techniques based on the statistics and linguistic knowledge help the business content expert to fulfill this task with more efficiency. Once the already identified activities were registered, the verb repartition statistics showed that only a few new activities are expected if a new business database is treated. That will keep the manual effort within acceptable bounds.

In order to register the activities according to the KB design, following steps are required for each activity:

- Choose a unique concept name for the activity.
- Provide a technical term used for transformation issues.
- Find out which activities can be synonyms with the current one.
- Determine whether the current activity is always the preferred form, the preferred form in specific contexts or always deprecated.

The activities are treated according to the occurrence statistics in order to optimize the business repositories coverage. Choosing a concept name is not always a piece of evidence. Words like 'prüfen' can be associated to the concept name 'TO_CHECK'. Some of the synonyms are clear: 'überprüfen' or 'checken' are interchangeable in their usage in the business expression denomination context. The simplest German variant 'prüfen' is the preferred form and the most used form in the occurrence statistics. The strategy for choosing a concept name is based on the English translation of the German term. Four questions must be answered to select the most appropriate English translation:

- How would the concerned business object denomination be translated in English?
- Is the concept name an appropriate technical term for transformation purposes?
- Does the translation fit to the context?
- Is the concept name as unambiguous as possible?

Translation software, translation web services or translation web pages like 'LEO' for the German language (*http://dict.leo.org*) can support the concept name choice. LEO provides very domain-unspecific translations, thus it proposes several terms. For example, the translations provided by LEO for 'prüfen' are in alphabetic order: 'to analyze', 'to assay', 'to audit', 'to canvass', 'to certify', 'to check', 'to consider', 'to control', 'to diagnose', 'to examine', 'to gauge', 'to inspect', to investigate', 'to overhaul'. Assuming that a German business object is called 'Bewerbungsunterlagen prüfen', it would be translated with the verb 'to check', which would also fit as a technical term. The verb 'prüfen' is a very general term, the other translations are more specific. In the insurances context, if the employee has to check whether the declarations of a person are correct, the term 'investigate' would fit better. But the concept 'TO_INVESTIGATE' better corresponds to the German term 'erkunden'. In this context, it would be more accurate to use 'TO_INVESTIGATE' than 'TO_CHECK'. When translating the synonym überprüfen with LEO, a large part of the previous translations are displayed along with some additional ones. When translating the verb 'checken', the unique translation 'to check' is proposed.

The previous example showed that interchangeable synonyms are existing in the business modeling context. However, the verbs significations depend much more than nouns from their context. For example, the verb 'kontrollieren' is a very close synonym of 'prüfen' and is often used instead of it, but in specific contexts, 'kontrollieren' is more appropriate than 'prüfen', for instance in controlling processes of a production line. The same case occurs with the previous insurance investigation problem. A first solution is to create as many concepts as required for each specific context, register all possible synonyms under them and define a different preferred form for each of them. The table 7.11 shows such an example applied to 'prüfen'. The header line contains some concepts, the first line displays the preferred form for each context and the following lines are deprecated synonyms.

TO_CHECK	**TO_CONTROL**	**TO_INVESTIGATE**	**TO_AUDIT**
prüfen	kontrollieren	erkunden	auditieren
überprüfen	prüfen	kontrollieren	kontrollieren
checken	überprüfen	prüfen	prüfen
kontrollieren	checken	überprüfen	überprüfen
		checken	checken

Table 7.11: Synonym relations between concepts

This example shows four different concepts. Each of them contains the best fitting term as preferred form. An interesting point is that more specific terms are synonyms of less specific terms: if someone needs the verb 'prüfen', he never will use 'erkunden' or 'auditieren' instead, but he could use 'kontrollieren', which is also common. If he wants to investigate or audit something, he may use a less specific term. It is why the specific verbs 'erkunden' and 'auditieren' are occurring under only one concept. If 'prüfen' is a synonym of an another term, this is also true for 'überprüfen' and 'checken'. This technique, even if precise, gives many redundancies in the KB. It is why an another strategy is preferred. Synonyms are grouped under one concept, forming a semantic cluster if they fulfill two conditions:

- They are interchangeable synonyms.
- One of the synonyms is never a preferred form in a similar context.

The remaining synonym relations are represented via the relation 'isContextSynonymOf' between two classes of the type 'FunctionConcept'. The table 7.12 shows the application of the previous rules to the example:

TO_CHECK	**TO_CONTROL**	**TO_INVESTIGATE**	**TO_AUDIT**
prüfen	kontrollieren	erkunden	auditieren
überprüfen	TO_CHECK	TO_CHECK	TO_CHECK
checken		TO_CONTROL	TO_CONTROL
TO_CONTROL			

Table 7.12: Synonym relations between concepts

The synonym relations are not represented, exclusively with relations between concepts, for two reasons:

- It would imply that each term is registered under its own concept. Each concept has an intrinsic semantics. Creating for example the concepts 'TO_CHECK_1', 'TO_CHECK_2', 'TO_CHECK_3' for 'prüfen', überprüfen' and 'checken' leads to concept redundancies and thus semantic redundancies which must be avoided.
- The natural languages constraints: a one to one translation for each word in each language does not exist. For example, the concept 'TO_CHECK' contains three items in German, but could have one or more items in other languages. In this case, a concept 'TO_CHECK_4' is required and no translations are available for the associated term in other languages.

This design choice requires that each concept has an accurate, unique and strong semantics. It is quite simple to define the difference between 'auditieren' and 'erkunden', but it is more difficult to define and limit the contexts where 'prüfen' and 'kontrollieren' must be used. It is why the registering process must be supported by methods which provide context information, the so-called context view. The mind map 7.15 summarizes which data must be provided by such a context view. In this example, the verb 'übergeben' is the current point of interest and thus the starting point of the mind map. The map is built as follows:

1. All business objects where the considered verb occurs are retrieved.
2. The nouns related to this verb are extracted from the business objects gathered in 1. This gives the list of all nouns which are associated to this verb in the corpus.
3. Retrieving all business objects where the extracted nouns occur.
4. All business objects where the nouns extracted in 3 occur are retrieved.
5. The verbs related to the latter nouns are extracted from the business objects gathered in 4. This gives the list of all verbs which are associated to the nouns associated to the starting verb in the corpus.

6. If available in the KB, the extracted verbs are associated to their corresponding concept(s).

The possibility to view the complete business object denominations associated to each noun/verb couple must be provided, too. The latter are not displayed in the map because of the lack of place. For the same reason, not all terms displayed in the map are expanded.

The synonyms detection is carried out visually, but can be supported by software or web services, which return a list of synonyms for a given verb. Not all synonyms are wished entries for the repository because some of them may never be used in business process models. The intersection between the returned synonyms and the verbs occurring in business repositories must be done. This set must then be analyzed in more detail with help of the context view in order to assign the verbs to existing or new concepts. The properly mapping of the terms to concepts can be supported by drag-and-drop functionalities.

The context view helps not only detecting synonyms, but also homonyms. If two or more significations of a term are used in the business context, they must all be registered under an appropriate concept. For example, the verb 'warten' has two different semantics and both are used in business object denominations, once in a maintenance context, the other time in the action of waiting for something. This verb is then registered under the two concepts 'TO_WAIT' and 'TO_MAINTAIN'. The context view will help to disambiguate which signification is meant. In the example map, the term 'übergeben' applies to the words 'Problem' and 'Eingangsbelege' (the two last relations at the left side). 'Eingangsbeleg' is also associated to 'weiterleiten'. If 'übergeben' associated to 'Eingangsbelege' means to give it to another person or organization unit for further processing, these two terms are synonymous in this context. Associated to the term 'Problem', 'übergeben' can also be a synonym of the verbs 'zuteilen' and 'zuordnen', which are very close synonyms as well. It is where crowdsourcing can really bring valuable content for taking such decisions because process modelers can contribute by explaining in detail why they chose the current term. For instance, 'weiterleiten' could be a term reserved for redirecting automatically some objects (virtual or real) via e-mail, software system or internal post service. The action 'übergeben' could be associated to manual effort, like an adaptation training for a certain topic.

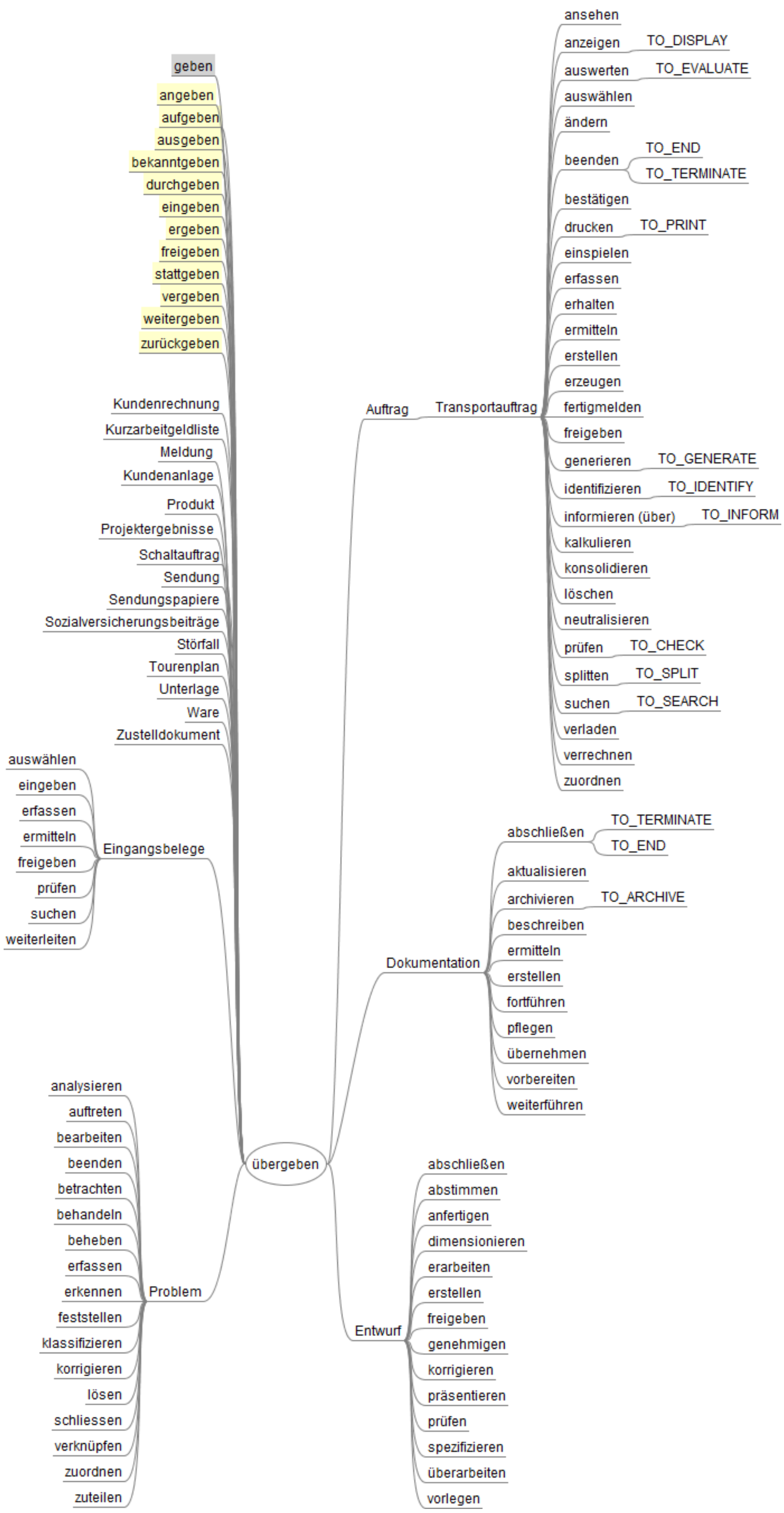


Figure 7.15: Relations of the activity 'übergeben'

The information highlighted in yellow is of different nature. In the example 'prüfen', one of the synonyms is 'überprüfen'. The two terms belong to the same grammatical family. They are both based on 'prüfen' and are differentiated only by the prefix 'über'. If some prefixes (detachable or not) are sometimes changing completely the signification of a verb, other prefixes/verb combinations can result in synonym terms or even enforce the verb semantics. As a support for detecting synonyms, all verbs issued from the same grammatical family are displayed. The LE is able to analyze such verbs and marks the position of the prefix, if available. In the example, 'übergeben' can be synonym of 'weitergeben' or only of the base form 'geben', which is very unspecific. Another typical example concerns the terms occurring in the UML diagram 7.14 which are assigned to the concept 'TO_SEND'. Two synonyms, 'senden' and 'schicken' are also synonyms with the prefixed terms 'versenden' and 'verschicken', which share the same prefix 'ver'. The term 'zurücksenden' has a different semantics (even if related) and can be associated to the concept 'TO_SEND-BACK'. The term 'zurückschicken' occurs as well in the corpus. Due to the synonym relation between 'senden' and 'schicken', one can assume that 'zurücksenden' is synonym of 'zurückschicken'. Only verbs which belong to the same family and which are occurring in the corpus are displayed in the context view.

To estimate the impact of prefixes on synonym possibilities, some statistics are performed on the verbs with prefixes that are occurring in the corpus. The verbs composed of a separable or non-separable prefix are grouped by the verb radical. Following prefixes were identified in the occurring verbs:

- Non-separable prefixes: ab, an, auf, aus, be-auf, bei, bekannt, bereit, dar, down, durch, ein, einzugs, entgegen, fehl, gegen, grob, fertig, fest, fort, frei, gegenüber, her, heran, herein, herbei , hin, hinzu, hoch, instand, mit, nach, nieder, rück, statt, sicher, still, teil, um, up, über, voll, vor, voran, voraus, weg, weiter, wieder, wieder-vor, zu, zurück, zurück-er, zusammen, zwischen
- Separable prefixes: an-er, auf-er, aus-be, be, bei-be, de, dis, durch, ein-be, ent , er, fort-ent, hinter, nach-be, nach-ver, neu-er, re, um, um-be, unter, über, ver, vor-be, vor-er, weiter-be, weiter-ent, weiter-ver, wider, wieder, wieder-ver, zer, zurück-ver

Some prefixes are composed of two prefixes. For more clearness, these are separated with a dash in the list. Some of the prefixes are likely synonyms, like 'fort' and 'weiter'. Others are related, like 'unter' and 'über', or 'aus' and 'ein'.

214 groups of verbs belonging to the same family were identified. In 123 groups, the radical form of the verb also occurs in the corpus. The largest group concerns the verb 'stellen', where the radical 'stellen' occurs 73 times in the corpus. The 17 verbs belonging to this group are (prefixes are highlighted by separating them with a dash): 'auf-stellen', 'aus-stellen', 'bereit-stellen', 'be-stellen', 'dar-stellen', 'ein-stellen', 'er-stellen', 'fertig-stellen', 'fest-stellen', 'frei-stellen', 'gegenüber-stellen', 'her-stellen', 'sicher-stellen', 'um-stellen', 'vor-stellen', 'zusammen-stellen', 'zu-stellen'. Such a

list can help identifying context-specific synonyms: in this example, 'fertigstellen', 'zusammenstellen', 'herstellen' and 'erstellen' can be synonyms depending on the context.

The described support for detecting synonyms differs from the one described in the use cases (document and business repository mining, term requests). The possible synonyms displayed in the cited use cases are directly issued from the KB because they base on already filled and reliable resources, whereas this section describes the first feeding of the KB with the corpus being the only available source of information.

Now that all the possible context information is available for taking the appropriate reification decisions the term can be registered in the KB. Following steps are performed automatically:

1. The concept name is adapted to fit the defined naming conventions.
2. A new concept class is created as a subclass of '*FunctionConcept*' if it does not already exist.
3. If the concept is a new class, the term is associated as a preferred form to this class, otherwise as a deprecated form.
4. If the class is a new class, the concept name is adapted to give a default technical term: the prefix 'TO_' is deleted and the remaining string is converted into lower case letters.
5. As a default value, the abbreviated form property is set to false.
6. Weight and lookup keys are automatically set according to the statistics and the output of the LE.
7. The identified context-specific synonym relations (supported by a GUI) are set.

The business content expert must verify and, if needed, correct the entry. After validation, the activity is registered in the KB.

7.3.2.2 Supporting Techniques for Registering Statuses

In the current design, four types of statuses were identified: results (statuses of type '*ResultConcept*', booleans (statuses of type '*BooleanConcept*'), modifiers (statuses of type '*ModifierConcept*' and colors (statuses of type '*ColorConcept*'). A different kind of support techniques must be provided for each category.

In events, verbs are conjugated in the past participle form when they are used as statuses. For example one of the possible result of the process step 'Kundenauftrag prüfen' is 'Kundenauftrag geprüft'. The LE is able to identify in which tense a verb is conjugated. A statistic process shows that from the 1017 different verbs which are occurring in the corpus, 604 are conjugated in the past participle form. No statistic is performed to see whether the corresponding infinitive forms are occurring in the corpus because the LE already delivers this information and the collected infinitive forms previously constituted the data set for registering activities. Registering again these verbs as a status would lead to redundancies in the knowledge base because the most attributes like the linguistic information, weight, synonym groups are the same. The only difference consists in the technical term attribute and the concept name. It is why the support for registering result statuses consists of automatically creating the link '*hasCorrespondingStatus*' between the classes of type '*FunctionConcept*' and '*ResultStatus*' and the opposite relation '*hasCorrespondingFunction*'. As a consequence, all attributes of the statuses are inherited from the function entries, including all synonym relations.

To avoid any human interaction in this process, a linguistic service returning the past participle form from a given English verb in the infinitive form can be coupled to this process. It will transform the technical name to a passive technical name and adapt the concept name to an equivalent passive form. For example, the concept 'TO_CHECK' is then related to an equivalent result status concept named 'CHECKED', whose technical term is 'checked'. It is not needed to register all synonym groups because all relations can be deduced from the existing data structure. For example, if the term 'überprüft' is used, the LE finds out that the infinitive form is 'überprüfen', which is associated as a deprecated form to the concept 'TO_CHECK', which in turn is related to the concept 'CHECKED'.

Adjectives and adverbs are treated in an identical manner. The registering process proposes the terms to be registered in the occurrence count order. A similar context view support must be provided as for the verbs: adjectives and adverbs are modifying either a noun ('vorhandenes Dokument', 'Ampel rot') or a verb ('manuell eintragen'). The context view shows all nouns and verbs associated to the adjectives and adverbs to help finding out synonyms. However, this category has fewer entries than verbs and the synonyms are well-defined: the context does not so often change the meaning of an adverb/adjective. The business content expert must choose whether the status is of type '*BooleanConcept*' or '*ModifierConcept*'. The LE can in some cases help taking this decision. A couple of adjectives/adverbs are available in their opposite form, like 'vollständig' and 'unvollständig'. The LE recognizes that the prefix ' un' or ' in' stands for a negated form. If both positive and negative forms are available in the corpus, the term is of type '*BooleanConcept*' and the relation 'hasAntonymBoolean' is automatically set. In the corpus, 34 adverbs and adjectives have their negative form available. The concept naming and attribute settings are running like for verbs in the same manner.

Finding out colors is based on a completely different support kind. The LE associates a semantic feature for each term. The semantic features classification is specific to each LE. Another LE can deliver different semantic values. For each LE, a mapping must be done between the values delivered by the LE and the classes structure which is common in all languages. In case of standard color adjectives/adverbs, a semantic feature called 'colour' is provided. Every time this attribute is encountered, the registering process proposes to create a color entry under the node '*ColorConcept*'. The business content expert must provide a concept name, which is also used as a technical term. However, the 'colour' feature is not only provided to the color adjectives/adverbs, but also to adjectives like 'farbig'. Due to this restriction, this process can only be performed semi-automatically. Only three colors are available in the corpus. Of course, external resources can be bound like external ontologies or morphological dictionaries of the LE to extract all standard colors at once (see chapter 9).

7.3.2.3 Supporting Techniques for Registering Nouns

Like demonstrated in the noun statistics, simple terms are mostly domain-unspecific. The major part of them must be registered in the standard knowledge base. Unlike during the verb registering activities, the knowledge base now contains some valuable information: the already stored verbs, adverbs and adjectives.

Nouns derive often from verbs and in the majority of cases, the LE is able to reduce related nouns and verbs to the same base form. A statistic check shows that 702 simple terms (of 2372) have a common base form with 503 verbs available in the corpus. These will serve as a starting point for registering almost a third of all simple terms semi-automatically. The concept name and technical name are adapted to nouns ('TO_' prefix removed) and synonym relations deduced and attributes are set.

If the entry is created automatically without any human interaction, the entry is marked with the '*entryStatus*' attribute set to '*generated*' in order to retrieve these entries for later validation. If the defaults are correct, the business content expert can validate them. Of course, manual changes must be possible for each attribute set by default. Once the entry was verified by the business content expert, the '*entryStatus*' attribute is set to '*validated*'.

According to the semantic features delivered by the German LE, the registering procedure can propose a classification of the noun concept within the knowledge base structure.

- 'process' and 'massnahme': this category characterizes mostly nouns that are deriving from verbs. It means that the noun denotes an action. 510 nouns are concerned. For example, 'Klassifizierung' is the action of 'klassifizieren'. These kind of nouns can be stored as child concept of the class '*ActionConcept*'. While this classification is correct in most of the cases, the business content expert must take care of some exceptions. For instance, 'Abteilung' is a business location and not the action of 'abteilen'. Specific relations (*hasCorrespondingFunction* and *hasCorrespondingAction*) relate the noun and the verb concept.

- 'agent&anto' and 'agent&ulo: 119 agents have one of these two semantic features which characterizes agents which are classified under the class '*AgentConcept*'.

- 'measure': 40 distinct terms can be potential units. The difficulty in this case is to distinguish real units from ad hoc abbreviations. Real units can be stored as child concepts of the class '*UnitConcept*'.
- 'time' and 'time&in': 19 distinct simple terms that denote some time spans are available, like 'woche', 'zeit', 'quartal'..., which can be stored as child concepts of the class '*TimeSpan*'.
- 'loc' and semantic features starting by loc (for example 'loc&city', 'loc&flaeche', 'loc&gebauede', 'loc&ag', 'loc&way'...): 162 simple terms, which must be taken with precaution. Some terms can be analyzed as proper names, which is not the right signification in the business context. The terms may be stored as child of the sub-categories of '*BusinessLocation'*.

Some words do not have a corresponding verb in the corpus (sometimes not existing). These remaining nouns cannot rely on pre-existing knowledge. A semi-automatic registering process cannot create a concept in this case, but can propose at least a classification according to the described semantic features. A context view is always required, to help the business content expert taking the right decisions. The classification per semantic features allows a better overview of possible synonyms: within the 119 agents, it is quite simple to find out that 'Abnehmer', 'Abholer', 'Besteller' and 'Kunde' are synonyms or at least related. Categories like countries, organization units or IT systems are mostly issued from different sources (see section 9).

The example 7.11 concerning the concept 'TO_CHECK' is reused to bring more clarity. Nouns having a common base form with the verb 'prüfen' (base form 'prüfen') are 'Prüfen', 'Prüfung' and 'Prüfbarkeit'.

The noun 'Prüfen' is an action deriving from the verb 'prüfen' characterized by the semantic feature 'process'. The noun 'Prüfung' is also an action deriving from the verb 'prüfen'. It is characterized by the derivation 'ung' and a semantic feature 'massnahme'. The semi-automatically registering procedure proposes following default attribute values and relations:

- The prefix 'TO_' is removed from the concept name 'TO_CHECK' so that the noun concept name is the remaining string 'CHECK'.
- The technical name is set to the concept name written with lowercase letters: 'check'.
- The technical plural name is set to the technical name and an additional 's': 'checks'.
- The usage form of 'Prüfung' is the preferred form, because the verb 'prüfen' is the preferred form of the concept 'TO_CHECK' and this noun occurs much more frequently than the spelling variant 'Prüfen', which is marked as a deprecated form.

- Both forms are not abbreviated.
- The linguistical attributes are always set automatically.
- The concept 'CHECK' will be stored by default as a child concept of '*Action-Concept*'.
- The relation '*hasCorrespondingAction*' is set between the concept 'CHECK' and 'TO_CHECK'.
- The relation '*hasCorrespondingObject*' is set between the concept 'TO_CHECK' and 'CHECK'.

A context view which displays relations with activities is always necessary. If the described registering procedure is correct in the case of 'Prüfung', it does not consider its homonym form, which means 'exam'. A second entry must then be created. Alternatively, the term can be associated to an already existing concept.

The noun 'Prüfbarkeit' is the capability concept of the verb 'prüfen'. It is characterized by the double suffix 'bar' and 'keit', delivered by the LE. The semi-automatic registering procedure proposes following attribute values and relations:

- The prefix 'TO_' is removed from the concept name 'TO_CHECK' and the suffix 'ABILITY' is added: the concept name is then 'CHECKABILITY'.
- The technical name is set to the concept name written with lowercase letters: 'checkability'.
- No technical plural name is required.
- The usage form is the preferred form because the verb 'prüfen' is the preferred form of the concept 'TO_CHECK'.
- This form is not abbreviated.
- The linguistical attributes are always set automatically.
- The concept 'CHECKABILITY' will be stored by default as a child concept of '*CapabilityConcept*'
- The relation '*hasCorrespondingFunction*' is set between the concept 'CHECKABILITY' and 'TO_CHECK'
- The relation '*hasCorrespondingCapability*' is set between the concept 'TO_CHECK' and 'CHECKABILITY'

The concept 'TO_CHECK' has two direct synonym forms: 'checken' and 'überprüfen'. To finish the entries concerning this semantic cluster, the nouns having a base form 'checken' and 'über$prüfen' are retrieved from the corpus. The nouns 'Check', 'Überprüfen' and 'Überprüfung' are found. They are registered as deprecated terms under the 'CHECK' concept like their corresponding verb form. This ensures a certain consistency between the verb form usage and the noun form usage.

As a last task, indirect synonyms corresponding to the verbs 'kontrollieren', 'erkunden' and 'auditieren' are searched. The verb 'kontrollieren' has the same base form as the nouns 'Kontrollieren' and 'Kontrolle'. A new concept named 'CONTROL' is proposed to store the noun 'Kontrolle' and 'Kontrollieren', where the most frequently occurring one of the two nouns is proposed as the preferred form. A synonym relation is set between the classes 'CONTROL' and 'CHECK'. The nouns 'Audit' and 'Auditierung' are available in the corpus, but only the base form of 'Auditierung' corresponds to the base form of the verb ('auditieren'). The noun 'Audit' has the base form 'Audit'. A fuzzy match against the verbs entries can be later done to find verbs starting by 'audit' when the noun 'Audit' will be registered. It can bring some false results, but in this case, 'auditieren' is the only matching result. No noun is found with the base form of 'erkunden'. The relations and attributes are set like in the previous procedures.

This synonym chain could be recursively followed until all corresponding noun forms are registered. If the business content expert performs the registering in an interactive mode, he can decide to follow the chain or to register the next term once the current cluster is completed.

In the previous example, the nouns 'Prüfen' and 'Prüfung can be used alternatively and thus are registered under the same concept. In contrary, the nouns 'Plan' and 'Planung' (the latter has the semantic feature 'massnahme') are both used in different concepts. 'Plan' would be stored under the concept 'PLAN' as child of the '*ObjectConcept*' class, 'Planung' under the concept 'PLANNING' as child of the '*ActionObject*'. The noun 'Planbarkeit' would be stored under the concept 'PLANNABILITY' as child of the class '*CapabilityConcept*'. The three nouns deriving from 'planen' are related to the concept 'TO_PLAN' via the relations '*hasCorrespondingObject*', '*hasCorrespondingAction*' and '*hasCorrespondingCapability*'. Note that the concept stored as an action has a name ending with 'ING', unlike the 'CHECK' concept. This is the default case when two of the variants are available.

The relation between noun concepts and verb concepts '*hasCorrespondingFunction*' is pointing to the verb form (*'FunctionConcept'*). No direct relation is available between the different object categories, like between objects and capabilities, to avoid redundancies and a cumbersome cross-relation handling. In case of a noun denoting an action or a capability, the verb form is always required: actions and capabilities are always deriving from a verb, so that it is in principle always possible to create automatically a verb entry, even if it does not occur in the corpus. The registering procedure is inverted. The verb form is then created in an identical manner from the noun form. This ensures that the cross-relations between nouns are available. These relations are necessary to detect a set of inconsistencies: 'Qualitätsprüfung durchführen' is equivalent to 'Qualität prüfen'. The quality kind should be more detailed: 'Dokumentationsqualität' or 'Produktionsqualität' for instance. Some objects may not have a corresponding verb form. In this case, no corresponding capability exists as well, so that no cross-relations between nouns are available.

Once the simple terms are registered, new valuable content is available in the knowledge base for registering compound terms. A statistical process cuts the parts of all compounds available in the corpus (5354) and checks them against all base forms that are available in the knowledge base at this point, those from verbs, adjectives, adverbs and simple terms. The results are:

- 3972 compounds have all parts available in the KB.
- 805 compounds have at least the head part (the last compound part) available in the KB.
- 448 compounds have some parts, excepting from the head available in the KB.
- 129 compounds have none of their parts registered in the KB.

These results show that three quarters of the entries can be registered automatically according to a systematical process. The entries can be validated on the fly or later by the business content expert. This process must be coupled with the method described above to classify the compound terms (general, domain-specific and company-specific terms). Independently to which domain a compound term belongs, the automatic registering process is based on two principles:

1. The compound is composed of the sum of its parts.
2. Each compound part specifies the other parts in more detail, starting from the head. For example, 'Lohndaten' is a subtype of 'Daten' and 'Bruttolohndaten' is a subtype of 'Lohndaten'.
3. When a concept belongs to a specific category, the sub-type is supposed to belong to the same category (Capability, Action, Agent, Object...). This assumption is often confirmed by the semantic feature delivered by the LE. For example, 'Verantwortlicher' is marked as an agent, 'Projektverantwortlicher' as well.

Of course, exceptions may occur: a word analyzed as a compound can in fact be a simple term, like 'Workshop' analyzed as 'Works' and 'hop'. This anglicism was not recognized as a common German word. Another category concerns abbreviated compound parts, like 'P-Auftrag'. In this case, it must be clear that 'P' is the abbreviated form of 'Produktion' and not some unit. However, in the majority of cases, the automatic registration will work successfully.

The following examples illustrate this procedure. Three terms related to the concept 'CHECK' are retrieved in the corpus: 'Qualitätsprüfung', 'Vertragsüberprüfung' and 'Auditprüfung'. The first one is composed of the words 'Qualität' and 'Prüfung'. If 'Qualität' is stored in the concept 'QUALITY', the automatic registering procedure will propose following attributes and relations:

- The concept name will be the concatenation of 'QUALITY' and 'CHECK', separated as a convention with an '_' : 'QUALITY_CHECK'.
- The technical name is 'qualityCheck', where the second compound part is written with an uppercase letter respecting general naming conventions in programming languages.
- The default technical plural name is 'qualityChecks'. Per default, only the compound head is set in plural form, which should already be registered in the simple term concept.
- The 'QUALITY_CHECK' concept will be stored as a child of '*ActionConcept*' like the 'CHECK' concept.
- A relation '*specifies*' is set from the 'QUALITY_CHECK' concept to the 'CHECK' concept and '*generalizes*' from the 'CHECK' concept to the 'QUALITY_CHECK' concept. Flat structures and relations are used instead of a subconcepts hierarchy because OWL DL does not allow classes to be also individuals.
- This term is the preferred form, because 'Qualität' and 'Prüfung' are both preferred forms of the concepts 'QUALITY' and 'CHECK'.
- The linguistic attributes are set automatically.

Analog to the previous example, the term 'Vertragsüberprüfung' will be stored in a new concept named 'CONTRACT_CHECK', but 'Vertragsüberprüfung' inherits from the deprecated form attribute of 'Überprüfung'. If 'Vertragsprüfung' is found in the corpus, it is proposed as the preferred form. Otherwise, the system alters the name 'Vertragsüberprüfung' by concatenating the compound parts 'Vertrags' and 'prüfung'. If synonyms of verbs or of simple terms have been carefully registered, the system already detects potential synonyms at this stage, like the third variant 'Vertragscheck'. This procedure is also used in the collaboration platform in the 'Request new term' use case (see figure 6.56). In the cited example, 'Bewerbungsgespräch' was already declared as the preferred form synonym of 'Interview', even if the number of compound parts differs. The proposed alternative consists in the concatenation of 'Bewerbungsgespräch' and 'fragenbogen'. The business content expert must always be able to correct the spelling of the generated term when some necessary grammatical elements are omitted, like the additional 's' between 'Vertrag' and 'prüfen'. In the latter example, the 's' is generated because the system has learnt it by taking example of the word 'Vertragsüberprüfung'.

Theoretically, all synonyms variants could be generated at this stage. However, this is not done because they are error-prone and would thus request at least a short validation from the business content expert to ensure the spelling correctness. It is why the synonyms are registered only if they are really used. On the other hand, because compound synonyms can be calculated, their registering could be omitted. This way is not chosen either because a calculated synonym may potentially mean something else. Once registered and validated, a simple look up ensures that the concerned term was previously validated and is the deprecated form. This is in particular important with context-specific synonyms: 'Prüfung' and 'Kontrolle' are related, but not always exchangeable. In the case 'Vertragsprüfung' and 'Vertragskontrolle', the business expert must either confirm that these terms are synonyms or two different objects. According to his decision, the concept 'CONTRACT_CONTROL' either is created or not. If not, the term 'Vertragskontrolle' is added as a deprecated synonym to the concept 'CONTRACT_CHECK'. In the opposite examples 'Qualitätskontrolle' and 'Qualitätscheck', the compound parts 'Check' and 'Kontrolle' are exchangeable synonyms. The concept 'QUALITY_CONTROL' is created, and both terms are stored as its children, the first as the preferred form, the second one as the deprecated one.

A last example found in the corpus ('Auditprüfung') is interesting because the compound parts 'Audit' and 'Prüfung' are context-specific synonyms. The original business object denomination is 'Auditprüfung durchführen', but the verb 'durchführen' must be avoided according to the guidelines. Replacing 'durchführen' with 'prüfen' ('Audit prüfen') is incorrect. The expression could be for instance 'Audit leisten', which is more precise. Anyway, terms with a redundant meaning like this one can be detected in this way and must be avoided. As a consequence, 'Auditprüfung' must be stored as the deprecated form under the simple term concept 'AUDIT'.

Compounds are not always a sum of noun terms: adjectives can also be involved like the term 'Neukunde' as opposite to 'Stammkunde', which is formed by the adjective concept 'NEW' and the noun concept 'CUSTOMER'. As new is stored as a status, potential inconsistencies like 'Neuer Kunde' and 'Neukunde' can be later detected.

Compound terms may also be written as a noun phrase like 'Art der Änderung' instead of 'Änderungsart'. A statistic process retrieving patterns composed in a sequence of noun, article, noun shows that 777 NPs are found in the corpus, 564 are distinct ones. NPs must be avoided and regular compounds must be used instead. The automatic registering procedure searches first whether an equivalent compound is available in the KB. The algorithm is the same as the algorithm used for the concept annotator of the SE-BPM applications (see section 8.3.1). If not found, the system proposes the concatenation of both nouns in the inverse order. If a compound including 'Änderung' is already available ('Änderungsgrad' for instance), the system learns that an additional 's' is necessary between the two compound parts. Otherwise, the business content expert must correct the term. A new concept 'CHANGE_TYPE' is then created, the compound form stored as the preferred form and the NP as the deprecated form.

7.3.2.4 Supporting Techniques for Registering Abbreviations

Business repositories are afflicted with ad hoc abbreviations and acronyms. The former are characterized by some lower case letters, mostly words that are unknown to the LE and finishing with a period. The latter are mostly written with some upper case letters. While some acronyms are well known, the ad hoc abbreviations are mostly initiated by business modelers who abbreviate common business terms to gain some modeling time.

Examples: 'durchf' or 'durchf.' for durchführen, 'erw' for 'erweitern', 'Pos' for 'Position'.

Abbreviations make the readiness decrease and prevent the system from automatically mapping abbreviations to terms if they are not registered in the KB. It is why abbreviations are forbidden in the QA-Model. Well-known acronyms (domain-specific or not) which are registered in the KB with their full-form are accepted, but their usage must be consistent. One of the two forms (long or abbreviated) should be deprecated.

However, many abbreviations are used in different databases. This demonstrates that modelers are intuitively using the same abbreviations for the same words. These abbreviations can be registered in the KB as abbreviated and deprecated form of a long form. To support the registering process, unknown words finishing with a point and other unknown words are matched with terms already registered in the KB. The matching process gives a list of terms starting with the same letters as the supposed abbreviation. The business content expert must decide according to the context to which concept the abbreviation belongs to. Acronyms often stand for first letters of compound nouns. For example, 'StVO' stands for 'Strassen-Verkehrs-Ordnung'. The registering process proposes compound terms matching the letters of acronyms. Of course, these two support mechanisms are not 100% reliable, but give satisfying results in the most common cases.

As a concluding remark, the KB serves as the authoritative resource for the SE-BPM applications. To ensure the correctness, the appropriateness and readiness of terms, the KB itself must respect the guidelines for object naming (see chapter 10). When generating automatically entries, the entry must be checked with the 'SE-BPM' applications (see chapter 8). If a term does not respect the rules, a warning must be generated and the business content expert decides whether the considered term makes an exception to the guidelines. No further warning will then be fired during modeling time. In particular, the NP and the hyphenation rules are concerned: hyphens must be avoided in compounds, unless they are composed of more than three parts or contain anglicisms. For example, if the term 'Qualitäts-Kontrolle' is proposed for registering, a hyphenation error must be generated.

7.4 Third Dimension: Aspect-Oriented Concept Classification

Added-value process chain models serve as entry points to navigate to the according business process models. In the use cases developed in the chapter 6, the information provided by the process map is used as an aspect-oriented classification of terms needed in several life cycle stages:

- Design: helps to classify process documentation in a given structure. That increases the documentation findability.

- Modeling: helps to choose and filter business terms in the controlled modeling use case (see figure 6.9).
- Execution: helps finding and annotating business processes.
- Governance: when new terms are created, they must be associated to a process map element (see figure 6.56).

As an initial step, the process maps stored in the different business databases from the corpus are exported. The occurring process map elements are associated to the business terms used in process models related to the considered process map element. As a result, half of the databases do not have any process map models, amongst these even some of the reference databases, which should reflect best modeling practices for different domains. The few existing process maps differ from each other. Some of them are very flat structured, others are incomplete and cover only a subset of modeled processes. The most complete one has six hierarchical levels, which are so detailed that they are extremely company-specific. At this stage, no real automation techniques are possible without building at least a common basic process map. The resulting statements are based on a manual examination of the exported data. Following examples were picked out of the exported process maps. They describe the naming structure of existing process map elements:

- The process map elements should denote process categories which are expressed by five explicit naming patterns:
  - Using explicitly 'prozesse' as the last compound part: 'Finanzierungsprozesse', 'Geschäftsunterstützungsprozesse'.
  - Using the verb 'durchfüren' or its substantive, 'Durchführung': 'Aktienhandel durchführen', 'Reisedurchführung'.
  - Using a noun denoting an action: 'Unternehmensführung', 'Fakturierung', 'Nachbearbeitung'.
  - Using a specific activity: 'Berechnungsformel festlegen'.
  - Using a substantive denoting an activity: 'Dokumentation Verrechnungspreisregelungen' where 'Dokumentation' stands for 'dokumentieren'.
- Implicit naming: nouns, where the missing compound part 'prozesse' can be easily understood: 'Beschaffung', 'Finanzwesen'.
- Unspecific naming: 'Finanzministerium', whatever the underlying processes could be, or 'Juni', denoting the time where the superior process map processes are executed, maybe with some differences between the months.
- Sometimes wild abbreviations are used: 'S/I-Vergleich Pers.kosten durchf.'.
- The noun is sometimes accompanied by an adjective: 'Vorbeugende Instandhaltungsabwicklung'.

- Process categories are in some cases coupled with a slash or an ampersand: 'Finanzen / Controlling', 'Finanzen & Controlling'.
- Hyphens serving as a name completion are sometimes used: 'Reise- / Kosten-controlling'.

As a general entry point, the process map element 'Geschäftsunterstützungsprozesse' (supporting processes category) and 'Unternehmensführung' (management processes) are the unique items found. Core processes are missing. Although it is difficult to find common map items in the examined corpus, several inconsistencies are available at the highest levels – essentially due to anglicisms and acronyms usage – which show that a common base process map makes sense:

- Personalwesen - Personalmanagement - Human Resources - HR
- Finanzwesen - Finanzen - Finance - FI
- Beschaffung - Procurement
- Travel - Reise
- Controlling - Kontrolle
- Risk process - Risk-Prozesse - Risikoprozesse

Fine-grained map elements that are deep in the hierarchy are mostly company-specific, like 'Datentransfer individuelle Ziele überwachen'.

These observations show that no standardized maps are available, even for the upmost levels. They are always newly created, sometimes based on domain-specific best-practices and can even be inexistent. A solution would be the development of a standard process map for the highest-level elements, respecting the naming rules that are specific to process map items. Such a map does not represent the organization structure, but a logical and a thematically navigation into processes. It provides then a set of content which avoids that the modeler has to start from scratch with his own added-value process chain diagram. If he wants to create his own one anyway, the logical structure elements can be reused and ordered differently while the standard map can furthermore be exploited for the controlled modeling use case and search purposes. The standard map can be extended by domain-specific map items, which can be developed in a collaborative way. The domain-specific map can itself be extended by company-specific items, mainly for the lowest levels.

The three entry points of a standard process map are: 'Management Processes' ('Führungsprozesse'), 'Core Processes' ('Kernprozesse'), 'Support Processes' ('Unterstützungsprozesse'). Even these top-level elements can suffer from naming inconsistencies. Underlying categories of support processes are for example 'Human Resources' ('Personalwesen') or 'Finance' ('Finanzwesen'). Such general process categories are available in each company. A second-level 'Travel Management' has in the most cases standard sub-processes like 'Travel expenses controlling' or 'Travel booking'. A standard map can be defined in such a way up to the third, in some cases the fourth level.

All created process map elements must follow the naming guidelines defined in the QA-model, which is developed according to the previous described pitfalls. Some of the rules are analog to these of functions and events:

- Correct structures are nouns denoting a process. They can be modified by an adjective.
- Map items defined in standard process maps must be preferred.
- The three top-level elements comprise the term 'Prozesse', so that it is no need to repeat it in underlying elements. This permits an increased readability of the map items.
- 'Durchführen' or 'Durchführung' is forbidden.
- Activity names must be avoided and substantives used instead.
- Anglicisms must be avoided.
- Abbreviations must be avoided.
- Term completion hyphens are forbidden.
- The coordinators 'and', '/', '&' must be avoided. In this case, the two items can be defined as children of a common upper map item.
- Unspecific items which do not denote processes must be avoided or used only as leaf elements (like month names). This point cannot be checked automatically with a high precision.

The KB structure must be extended to allow the registering of the process map items. A new class is created, named '*ProcessMap*'. Each process map element is represented by a unique concept stored as a subclass of '*ProcessMap*'. As a convention, map concepts are starting with the string '_MAP_' to distinguish them from other concepts. The top-level elements are named '_MAP_CORE_PROCESSES', '_MAP_SUPPORTING_PROCESSES' and '_MAP_MANAGEMENT_PROCESSES'. The map concepts are related with following relations:

- *hasParentMap*: relates the process map element to its upper-level process map element
- *hasChildMap*: relates the process map element to its lower-level process map elements

Some process map elements, like 'Change Management' or 'Project Management', may occur several times in the process map because change or project management can concern different process types. In such cases, the concept name must be duplicated and extended with a numerical identifier to prevent from mixing parent and child elements and thus avoid the creation of a graph instead of a process tree.

Each map concept stores several individuals, which represent the different possible namings for the map element, except from pure spelling variants which can be detected automatically. As each individual naming can be composed of an expression containing several words, the individual names reflect per convention the concept name, for instance '_map_core_processes_n', where 'n' is an integer incremented for each individual. Each individual contains following attributes:

- *citationForm*: string, denoting the name of the element
- *usageForm*: of type '*UsageForm*', denotes whether the individual is the preferred, accepted or deprecated form.
- *lookupKey*: string, containing the concatenation of the base forms of the whole name, to allow the matching of the morphological analyzed process map elements against this string. This allows also the mapping of spelling variants (mainly hyphenation variants).

Attributes like weight, abbreviated form or technical names which are available for other kind of concepts are not necessary here because no use case makes usage of them.

The governance level is the best phase to define the needed process map elements. Elements from the standard map can be enabled or disabled according to the modeling needs (and in this way define the modeling scope), imported in the modeling environment and translated in added-value process chain models. To maximize the benefits of the aspect-oriented classification of terms in the different life cycle stages, terms must be mapped to process map elements. The previously exported process maps contain the terms belonging to each process map element. These terms are nouns which appear in business objects related via their container model to the process map elements. A manual examination permits following observations:

- Unspecific terms like 'Daten', 'Belege', 'Info' or 'Unterlage' are appearing in many process map elements.
- Some more specific, yet common terms like 'Kundendaten' can appear in several process map elements.
- Very specific terms are appearing in only few or even in only one process map element.
- Terms related to a given process map element are often, but not exclusively, related, like following examples:

  - Risk management: 'Risikoinventur', 'Risikobearbeitung', 'Risiken', 'Risikohandbuch', 'Frühwarnsystem', 'Frühwarnindikatoren', 'Sollwerte' , 'Toleranzgrenzen'
  - Travel management: 'Reisen', 'Reiseart', 'Reiseoriginal', 'Reisenbelege', 'Belege', 'Belegreport', 'Auslandsbelege', 'Inlandsbelege', 'Posten', 'Position', 'CD'
  - Change management: 'Testfallspezifikation', 'Testfälle', 'Testaktivitäten', 'Testmerkmale', 'Smoke-Test', 'Beta-Test', 'Beta-Tester', 'Testplanung ', 'Validierung', 'Version', 'Fachkonzept', 'Implementierung'

- Terms can be associated to several process map elements in the lowest levels, but they mostly have a common parent process map element.

From the previous observations, we can conclude that is it worth to associate terms to concept map elements to fulfill the requirements of the different use cases. The process map can then even act as a context provider when unspecific terms are used. For example, if the term 'Test' is used in a process model associated to the change management process map element, according to the example above, the preferred propositions can be 'Smoke-Test' or 'Beta-Test'. Following statements are concluded for the classification of concepts in the map:

- Unspecific terms like 'Daten' are not classified in the process map. These terms are mostly simple terms associated to many more specific compound terms like 'Kundendaten'. The term 'Bewerber' can be placed under the map element 'Personalbeschaffung' because the only more specific term is 'Initiativbewerber'.

- It is wished that a concept belongs only to one process map element. However, if a concept occurs oftentimes in several map elements, exceptions can be made to favor the term findability in these contexts. In contrary, if a concept only occurs a few time in given process map elements and oftentimes in an other, it will be associated to the element where it occurs the most. Example: a 'Bewerber' will always intuitively be searched under 'Human Resources - Personal Procurement' ('Personalbeschaffung'), even if it is needed in other process types.

- It is rarely worth to associate terms to the lowest process map elements. The fourth level will probably be the lowest one, unless the sub-processes are using very specific objects. For example, leaf nodes like 'June' or 'September' are probably using the same terms, even if the processes are different.

This basic statistic and the conclusions above allow the mapping of some of the terms to process map elements, but not all due to the lack of high quality data. When a new term is requested (see figure 6.51), the current modeling context is taken (where the process model is created) as the default process map element. This location can be changed manually by the business context expert.

From a data model point of view, two new relations must be created between subclasses of '*AbstractObjectConcept*' and subclasses of '*ProcessMap*':

- *belongsToProcessMap*: relates '*AbstractObjectConcept*' to '*ProcessMap*'
- *containsObject*: relates '*ProcessMap*' to '*AbstractObjectConcept*'

7.5 The Knowledge Matrix

In previous sections, a data model for storing business terms in the KB was developed according to the interpretation of statistical analysis of business repositories. The data model results in a set of classes, static and dynamic relations between single terms or noun phrases. Although business object denominations are sometimes formed of a single term (standard case for technical terms), they are more often composed of several terms like the association of a noun and a verb in function objects. This composition is up to now called 'Business expression'. The current data model of the KB does not reflect complete business expressions. The knowledge matrix (KM) is an extension of the data model, which serves as a storage and lookup of complete business expressions including the semantic layer on conceptual level.

The KM must be able to cover one to several dimensions. The unidimensional matrix entries cover cases like technical terms that are usually composed of a single noun and serve as a learning base to build automatically other dimensions.

7.5.1 Unidimensional Knowledge Matrix

Each concept which is a subclass of '*AbstractBusinessConcept*' can potentially occur alone in business objects. If nouns are occurring alone, they are supposed to be in singular form. For example, while the noun 'Bewerbungsunterlage' is mainly used as a plural form in a function object ('Bewerbungsunterlagen prüfen'), it remains in singular when modeled as a technical term object. Adjectives, here treated like statuses, are per default in singular. Verbs, if they are considered as activity names, are used in the infinitive form. If the status form is meant, they are used in the past-particle form. Nouns, verbs, adverbs and adjectives are forming the first matrix dimension.

To store the matrix elements, the data model is extended with a new top-level class called '*BusinessExpression*'. Three subclasses are needed to describe the unidimensional entries:

- *BObject*: contains noun entries, denoting business objects, independently of their specific semantic category, which is already represented in the data model.
- *BActivity*: contains verb entries, used as activity names in the infinitive form.
- *BStatus*: contains verbs conjugated in the past participle form, adjectives and adverbs.

Each business expression is represented by concept of its own. To avoid naming collisions with the term concept names, the business expression concept name is as convention composed of the concatenation of the prefix '_BE_' and of the considered concept name, like for example '_BE_DOCUMENT'. Business expression concept classes contain one children individual per language. The individual name is the same as the concept name, but written with lower case letters. If the language-specific resources are placed in different files, no extension is needed to ensure name uniqueness. The underlying individuals contain two attributes: '*entryStatus*' of type '*EntryStatus*' and a string named '*businessExpressionString*'. The latter attribute contains the business expression which is used in business objects. These entries are created automatically, but the business expression string can be manually corrected. The corrected form will be the authoritative one. Following steps are performed to create the unidimensional entries:

- For every concept, the preferred form individual is retrieved. Deprecated or accepted forms are ignored.
- The citation form provided by the LE, which is stored in the considered individual, is taken as the default business expression string. It is supposed to follow the correct hyphenation rules and is always in the singular form, infinitive form or singular masculine form for nouns, verbs and adjectives respectively.
- An exception is made with verb statuses. The business expression string consists there in the verb conjugated in the past participle form ('geprüft', 'informiert'). This form is provided by the corpus, by business objects created in a restricted modeling use case or is requested by a business modeler in a controlled modeling use case.

In the German language, the past participle remains invariant, but in some other languages like French, it varies according to the object. Then the masculine singular form must be taken. While the base procedure (classes, naming conventions and attributes) is the same for each language, the building of business expression strings must be adapted.

The fact that only preferred forms are registered as business expressions leads to a set of correct business expressions. Anyway, unspecific terms make an exception: terms like 'Data', even if they are preferred terms, are too unspecific. To cover this case, a parallel structure is created, with a top-level class called '*NegativeBusinessExpression*'. It contains the same subclasses as the class '*BusinessExpression*', but prefixed with the letter 'N', which stands for 'negative' ('*NBObject*', '*NBActivity*', '*NBStatus*'). As default, nouns associated to more than three specific compounds are considered to be unspecific and are stored in a same manner in the negative hierarchy. Otherwise, they are supposed to be specific enough. The business content expert can move manually the nouns from one hierarchy to the other. The concept which stores the verb 'durchführen' is also moved to the negative business expressions: it is considered to be too unspecific.

7.5.1.1 Multidimensional Matrix

Once the unidimensional matrix is built, the challenge is to fill a multidimensional matrix with business expressions. The association of an object and a verb results in a two-dimensional matrix. The association of an adjective, an object and a verb results in a three-dimensional matrix. The association of an adjective, an object, an adjective and a verb results in a four-dimensional matrix and so on. The dynamic relations defined in the table 7.10 describe usage constraints that applies to the business expressions.

Multidimensional concepts can be filled during several use cases:

- After business repository mining: when business expressions are retrieved and recognized as a correct structure by the SE-BPM applications, they can be stored as correct business expression strings. The '*entryStatus*' is set to '*generated*'. The business content expert can validate them later.
- During a controlled modeling use case: in the wizard, the business modeler associates objects to activities, to modifiers or to statuses. If not already stored in the KM, the business expression string is generated (explained later in the current section). If the business modeler does not mark the result as incorrect, the generated string is stored in the KM.
- During a restricted modeling use case: after the business modeler has created the business expression, the system interprets it and marks it as correct or incorrect. If the structure is correct, the business expression can be stored in the matrix and corresponding dynamic relations are created. The business content expert can validate the created entries. This prepares the switching from restricted modeling to controlled modeling.
- During governance time: if new relations are validated between the different business terms, the business expression string could be generated and verified at the same time. This avoids further checks by the business modeler, but increases the effort of the business content expert.

The simplest structure of a multidimensional business expression is the association of a business term with an activity (standard naming rule for process steps). A new subclass of '*BusinessExpression*' called '*BObjectBActivity*' stores business expressions sharing this special structure. Each business expression is stored in a concept, named '_BE_ObjectConceptName__FunctionConceptName'. The object and the activity concept names are separated through two underscores to differentiate them from the underscores which can already be contained in the object or activity concept name. For example, the business expression string 'Kundenangebot prüfen' is stored as an individual under the concept name '_BE_CUSTOMER_OFFER__TO_CHECK'.

Concept entries of the multidimensional matrix have two attributes: '*technicalName*' and '*technicalPluralName*'. The technical name attributes are formed as a concatenation in a given order of the technical names and technical plural names stored in the related concepts. More details about the technical name generator will be given in the section 8.3.6. These technical names can be manually adapted by the business content expert if they are incorrect. The corrected versions will be the authoritative ones. The plural version is necessary because some business expressions may be used in plural. When possible, the plural forms must be avoided and modeled as a loop for instance. However, in some cases it is not possible. For example, in high level processes a function called 'Reisebelege prüfen' may be modeled. This function is named in the plural form, but can serve as an entry point to a sub-process which details how to check the travel expenses. For the same reason, individuals stored under multidimensional business expression concepts come with two string attributes: '*businessExpressionString*' and '*businessExpressionPluralString'*. Only one of the two attributes is required, according to the concrete usage and KM filling strategy. The LE recognizes whether the string is used in the singular or plural form. The '*entryStatus*' attribute is always available, like for unidimensional business expressions.

Some business expression structures involving a business status can be expressed in the negated form like 'Bewerbungsunterlage nicht vorhanden'. To solve this problem, the concerned classes have two additional attributes called '*businessExpressionNegativeString*' and '*businessExpressionNegativePluralString*'. These entries will be constructed with an analog method as for the plural entries.

The following subclasses of '*BusinessExpression*' describe some additional standard business expression structures:

- *BObjectBStatus*: relates an object to a status, like 'Reisebelege geprüft' or 'Reisebelege vorhanden'.
- *BObjectModifierBObjectBStatus*: relates an object to a status and a modifier, like 'fehlende Reisebelege vorhanden'.
- *BObjectBStatusModifierBStatus*: relates an object to a status and a status to a modifier, like 'Reisebelege manuell geprüft'.
- *BObjectModifierBObjectBStatusModifierBStatus*: relates an object to a status and a modifier, and the status to a modifier, like 'fehlende Reisebelege manuell geprüft'.

This results in a list of correct business expression structures. This list is not exhaustive and can be extended if needed, but with respect of business object naming guidelines. Some structures need the usage of a placeholder string, for instance when literals, numbers or IT systems are used. Examples:

- *BObjectBLiteralBActivity*: stores business expression strings like 'Meldung "Server nicht vorhanden" anzeigen'. This string would be associated to a concept named '_BE_MESSAGE_%LITERAL_DISPLAY'. The business expression string contains then a variable called '%LITERAL' ('Meldung %LITERAL anzeigen'). The allowed literal list can be stored in an external resource.
- *BComparatorBNumericalBUnit*: stores business expressions like 'kleiner 5 km'. This string would be associated to a concept named '_BE_LT_%NUMERIC_KM' and the business expression would be 'kleiner %NUMERIC km'.

- *BObjectBActivity*: can store a concept named '_BE_%ITSYSTEM__UPDATE' associated to a business expression string named '%ITSYSTEM aktualisieren'. The IT system variable can be replaced by IT system names stored in the '*ITSystemConcept*' class.

Parallel to the multidimensional classes, the same subclass structure prefixed by 'N' for 'negative' is stored in the class '*NBusinessExpression*'. These negative classes do not contain unwished business expression structures, which are filtered out by the SE-BPM applications (see section 8.3.2). They contain business expressions which have a correct structure, but describe some vocabulary restrictions. Three cases can occur:

1. The business modeler requests new relations between terms, like a new relation between an object and an activity. The relation is refused by the business content expert. Such an association is then stored as a concept in the corresponding subclass. For example 'Kundenangebot lesen' would be stored under the class '*NBObjectBFunction*' as a concept '_BE_CUSTOMER_OFFER__READ'. No individual is required in this case. Such a concept creation prevents from a subsequent identical relation request. A link to the positive formulation can be set. This relation, called '*preferredExpression*', points to a positive business expression concept.

2. A set of accepted relations between objects, activities, modifiers and statuses are defined. In some cases, the association of all these objects could be incorrect. For example, 'Reisebelege automatisch prüfen' is correct, but 'Fehlende Reisebelege automatisch prüfen' could be incorrect. Such an association would be stored as a concept under the corresponding negative subclass, analog to the first case.

3. In a restricted modeling use case, the business modeler may use an unwished business expression. The system makes then a proposition. If the proposition is accepted by the business modeler, the unwished business expression can be stored as a concept under the negative business expression hierarchy with a link to the positive business expression concept. Example: 'Benutzeranmeldung durchführen' must be replaced by 'Benutzer anmelden'. The next time the same false expression is used, it can be automatically replaced by the correct proposition. This procedure cannot apply to unspecific nouns because the correct expression is then ambiguous.

7.5.2 Multidimensional KM Entries Generation

In most of the cases, business expressions are only stored in the KM if they were previously used. In the standard KB, the business expression entries grow on a collaborative way and are then available for the modeler community. If a needed business expression – for instance in the controlled modeling wizard – is not already part of the KM, but composed of registered concepts, it is built on the fly. The building procedure must be adapted for every language. This section describes some ground building mechanisms for the most common business expression structures for the German language.

Following steps apply for the generation of each kind of business expression strings:

1. The ground principle is the concatenation of existing business expressions or parts of existing business expressions.

2. A new concept is created according to the naming conventions described in the last section. These concepts cannot be created manually to prevent from naming errors and thus to ensure the correct retrieval of business expressions based on the concept names.

3. An individual is created according to the naming conventions. The generated string is stored under the attribute '*businessExpressionString*' or '*businessExpressionPluralString* according to the singular or plural usage.

4. The attribute '*entryStatus*' of the individual is set to '*generated*'.

5. The business expression is proposed to the business modeler who can mark it as correct or incorrect.

6. The business content expert validates the business expression and corrects it if needed. The '*entryStatus*' is set to '*validated*'.

7. If the business expression was corrected, it can be replaced with the correct form in the business repository.

Concerning the plural form or the flection of adjectives associated to a noun (number, gender and case), two different procedures can be adopted:

- Using a linguistic generation software, which takes the adjective or the noun as an input parameter with the wished flection attributes: plural for nouns, gender, number and case for adjectives. The drawbacks are additional software costs and sometimes a lack of reliability if the terms are unknown to the LE.

- Using already generated business expressions as an example to generate new ones. This has the advantage of using only local resources. If no appropriate example is available, the wished business expression can be created in a collaborative way and further on serve as an example to create analog expressions.

The business expression string for the classical process step structure – association of a noun or a noun phrase with an activity – consists of the concatenation of:

- The attribute '*businessExpressionString*' of the unidimensional business expression of the noun concept (element of the '*BObject*' class)

- A blank

- The attribute '*businessExpressionString*' of the activity concept (element of the '*BActivity*' class)

If the generated business expression is needed in the plural form, the system retrieves other business expression strings having the same structure and using the same concept. If one of the retrieved business expressions has the attribute '*businessExpressionPluralString*' filled, the plural noun part is picked out and associated to the activity. To allow the retrieving of a part of business expressions, each part must be separated from the others. To ensure that, a marker (deleted before usage) is placed in the business expression string, like following example:

%BObject Bewerbungsunterlage %BActivity prüfen

These markers are required for two reasons: firstly, a concept does not always stand as one word, especially in non-compositional languages, where compound terms are noun phrases. Analog to that, verbs can have separable prefixes. Secondly, depending on the language, the business expression structure is not the same. In function names written in English, the verb comes first, followed by the noun.

The business expression string for the classical event object structure – association of a noun or a noun phrase with a status – consists of the concatenation of:

- The attribute '*businessExpressionString*' of the unidimensional business expression of the noun concept (element of the '*BObject*' class)
- A blank
- The attribute '*businessExpressionString*' of the status concept (element of the '*BStatus*' class)

The business expression string for the function object structure 'object modifier + object + activity' consists of the concatenation of:

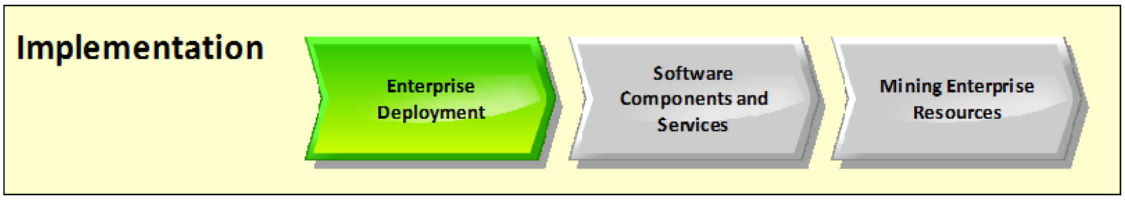
- The adjective inflected according to the following noun. Business expression strings, using the same adjective, associated to a noun having the same gender and number are retrieved. The part marked with '%ObjectModifier' is picked out.
- A blank
- The attribute '*businessExpressionString*' of the unidimensional business expression of the noun concept (element of the '*BObject*' class)
- A blank
- The attribute '*businessExpressionString*' of the activity concept (element of the '*BActivity*' class)

Each business expression structure needs its own language-specific building rules. Collaborative work can help correct generation errors. The resulting KM as well as the underlying KB serve as the main resources for SE-BPM applications, which are detailed in the next chapter.

Chapter 8

Enriched Life Cycles and Business Repository Interactions

The quality management framework is enhanced with software components which control the quality of the business objects stored in the business repository. The knowledge matrix and the underlying knowledge bases serve as the main knowledge back-end used by a set of syntactical and semantic annotators which ensures the quality metrics of business object labels. Producer annotators (the SE-BPM engine) supplement the label with information later exploited by consumer annotators and applications. This method provides control and quality assurance of the natural language input during the modeling activity in terms of appropriateness, correctness, consistency and translation readiness, as well as the fundament for support functionalities in all stages of the BPM life cycle. The SE-BPM engine gains information from the governance level and after applying reasoning algorithms, enriches and supports BPM governance with additional information.



8.1 Solution Overview

The solution is designed to be deployed as a service in a Service Oriented Architecture. This service is runnable in workflow engines provided by business process modeling tools in order to enrich the Business Process Management life cycle and to integrate semantic facilities in the core of the process management activities.

The figure '8.1' shows the deployment and the interactions of the SE-BPM engine. The information provided on governance level (terminology, compliance rules) are integrated in the company-specific SE-BPM resources (KB and KM) by the business content expert. The SE-BPM applications exploit them at runtime for controlling and assuring the quality of natural language input on the business modeling level, ensuring a consistent business repository that empowers subsequent process management tasks within and outside the modeling environment.

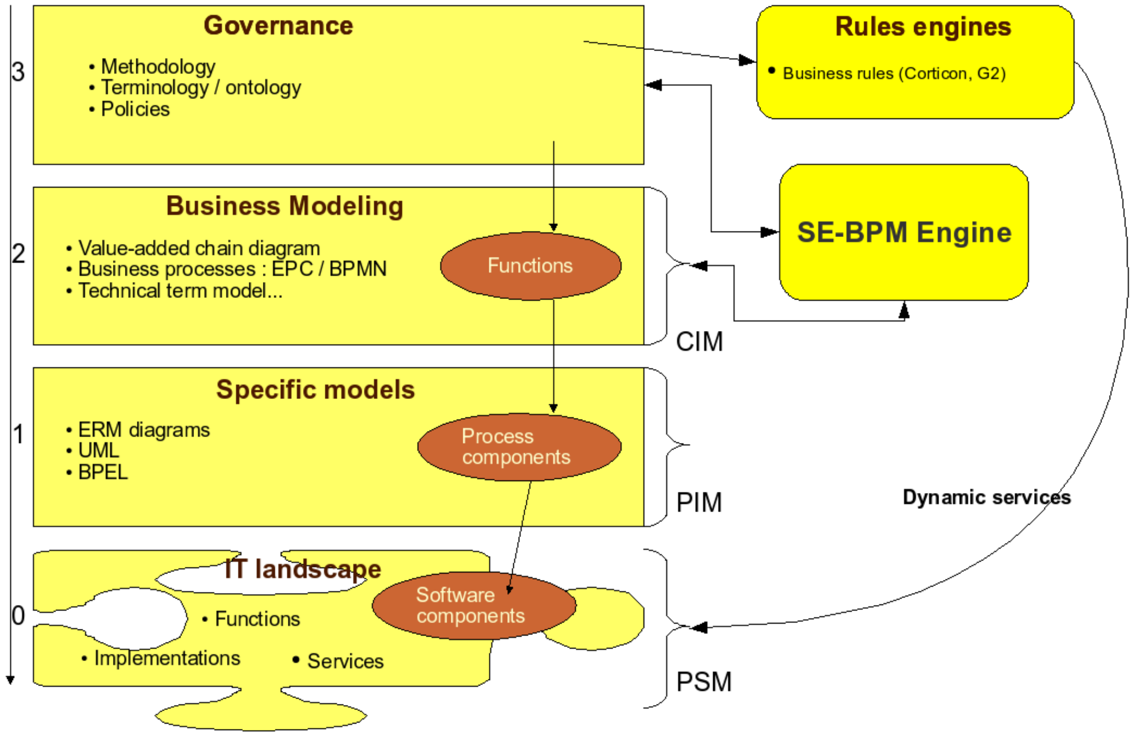


Figure 8.1: SE-BPM deployment

The figure 8.2 shows an overview of the main components of SE-BPM. Each of them can be deployed as a software service. The services can be orchestrated with a workflow formalism like BPMN or BPEL according to specific flows determined on the governance level. The 'SE-BPM pre-processing' component contains a set of pre-processing sub-components handling the rough data (business expressions). The 'SE-BPM core' component contains a set of producer annotators basing on the output of the pre-process. The 'SE-BPM applications' component contains a set of consumer applications which exploit the information provided by the produced annotators and prepare the output in a user-friendly way.

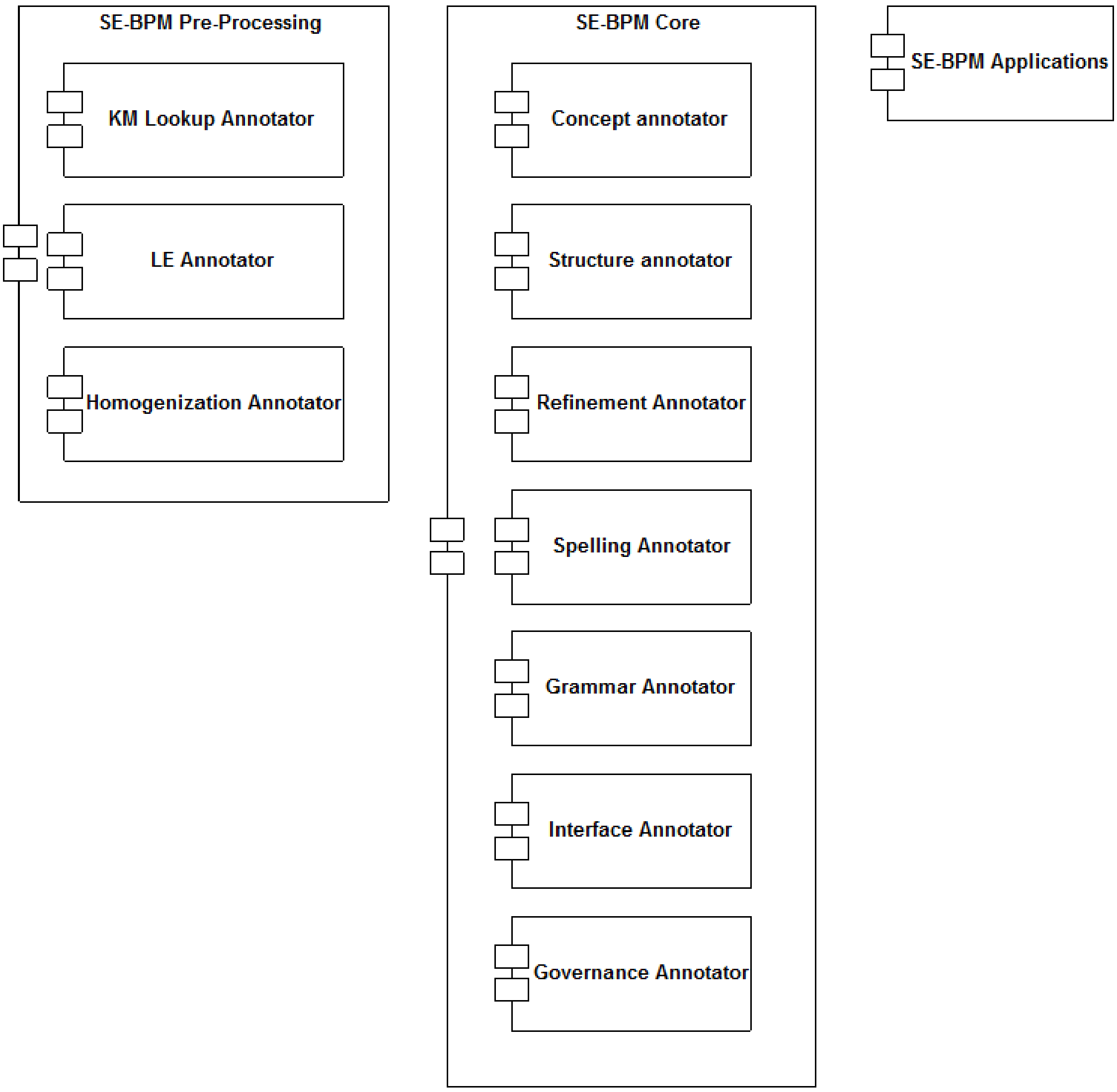
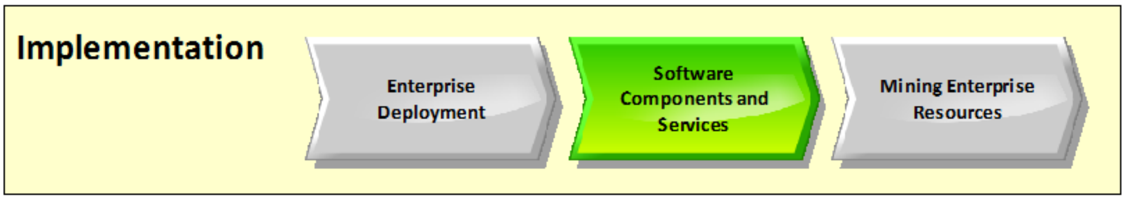


Figure 8.2: SE-BPM main components



8.2 SE-BPM Pre-processing Components

The pre-processing component is the first step of the annotator chain. It performs basic operations on business expressions. Following input parameters are needed during the whole process chain within the SE-BPM engine:

- The business object denomination.
- The unique identifier which permits the mapping between the produced results and the business object stored in the business repository.
- The object type to configure some of the processing steps. At the current development stage, the object types are: event ('TYPE_EVENT'), process step ('TYPE_FUNCTION'), data object ('TYPE_DATA') and process map element ('TYPE_PROCESS_MAP'). The listed type names are abstract names which are used internally to select operations adapted to each specific business object type. Since the type names do not correspond to the names chosen by BPM tool vendors, a mapping must be performed between the object types provided by the modeling environment and the type names used in the SE-BPM engine.

8.2.1 The KM Lookup Annotator

The first of all annotators is actually a bypass process. The treated business expression is searched as a whole string in the KM against the attributes 'businessExpression-String' and 'businessExpressionPluralString'. If the business expression matches, the lookup annotator provides additional information about the string:

- If the match concerns a correct business expression, the annotator returns:
  - The semantic layer: the list of concepts involved in the business expression, which can be exploited in subsequent BPM tasks like process retrieval or merging activities.
  - The technical name: can be used in case of a transformation in an IT solution
- If the match concerns an incorrect business expression, the annotator returns:
  - If available, the link to the correct business expression with all previously cited information.

– If not available, the error code like the usage of unspecific terms or forbidden relations.

If this annotator provides a result, the annotator chain is interrupted and the results are returned to the calling instance. As the objective of the whole annotator chain is to bring exactly the same result, a more detailed analysis of this business expression is unnecessary.

8.2.2 The LE Annotator

The morphological annotator (LE) is used for tokenizing and assigning generic linguistic information to words. A chunking pre-process serializes the input into meta-information about the processed resource and LE-relevant information. See 4.5 for more details about the LE, which operates as an enabler for further annotators.

The business expression is chunked into word tokens. Each word is annotated with following linguistical features issued from the German LE and used for further processing:

- *ori*: the original form of the word.
- *c*: the grammatical main category of the word, mainly nouns, verbs, verb prefixes, adverbs, adjectives, articles or numbers. For function words, the category has the attribute 'w' and the feature 'sc' describes it more precisely.
- *sc*: the grammatical sub-category, used in combination 'c=w'.
- *vtyp*, *nb*, *per*, *tns*: grammatical features describing the verb form (type, number, person and tense).
- *nb*: grammatical number. If the number is set to plural for the business object, the business expression is considered to be in plural.
- *lu*: the lexical unit, which is the stem form of the word.
- *ls*: the lexical structure.
- *zf*: the citation form ('Zitatform').
- *ts*: describes the basic decomposition of a compound word with a hash-separated value. For instance, the word 'Bestelldaten' has the 'ts' value 'bestell#daten'.
- *s*: semantic annotation anchored in the morphological lexicons, like 'loc&city', 'month' or 'money'.
- *state*: contains the value 'unknown' if the LE did not recognize the word. This feature also reports inadequate hyphenations.

8.2.3 The Homogenization Annotator

The LE is language-specific. Since no LE can process all languages, different kinds of LE may be integrated in the SE-BPM engine. While the information required from the different LEs is similar for each language, the output format can differ. The homogenization annotator standardizes the output of the different LE in a common representation that comprises the information necessary for further processing in a uniform way.

The different word tokens are marked in the business expression. Each word is annotated with following attributes:

- *id*: the business object id, to allow further mapping between the processing results and the business object stored in the business repository.
- *category*: grammatical category. The category ('c') and sub-category ('cs') issued from the German LE are mapped to this unique category attribute. The main values are nouns (NOUN), verbs (VERB), adverbs (ADVERB), adjectives (ADJECTIVE), articles (ARTICLE), coordinations (COORD) and others (OTHER).
- *verbForm*: maps the 'vtyp', 'nb', 'tns' ,'per' delivered by the LE to one of the following verb forms: 'INFINITIVE', 'PAST_PARTICIPLE', 'CONJUGATED'.
- *plural*: if the number feature associated to nouns has the value 'plu' (plural), the boolean attribute plural is set to 'true'.
- *citationForm*: value provided in German by the 'zf' feature. Serves as a default citation form if the word has not already been stored in the KB.
- *lookupKey*: value provided in German by the 'ts' feature. Serves to match the 'lookupKey' of concepts stored in the KB.
- *basicLookupKey*: value provided in German by the 'ls' feature. Serves to match the 'baseLookupKey' of concepts stored in the KB.
- *variantLookupKey*: value provided in german by the 'lu' feature. Serves to match the 'variantLookupKey' of concepts stored in the KB.
- *semantics*: open string values which may differ for each languages. These differences are not very important since the rules to handle them are also language-specific.
- *unknown*: boolean value, set to true if the word was not recognized by the LE.

Once these attributes are filled, the annotated structure can be further processed by the core components of the SE-BPM engine.

8.3 SE-BPM Core Components

8.3.1 The Concept Annotator

The purpose of this section is to describe the mapping between the natural language terms – word chunks identified during the pre-processing steps – and the language-independent concepts issued from the KB. The concept annotator is the first one of the core components because the information collected from the KB may refine the results of the LE. Words marked as unknown by the LE may be well-known in the KB, like the usage of business-specific anglicisms, organization unit names or IT system names. The concept mapping adds a semantic layer on top of the business expressions.

The concept annotator tries to match the stem form of a business term with the 'lookupKey' attribute associated to individuals. If one or several matches are found, the term is annotated with the concept(s) to which the individual belongs.

For example, the business expression 'neues Fahrzeug auf Lager' results in following concepts and parent class associations:

Fahrzeug	->	VEHICLE	–>	`ObjectConcept`
neu	->	NEW	–>	`ModifierConcept`
Lager	->	STOCK	–>	`LocationConcept`

In the KB, the lookup keys can be the same for several concept categories. In languages like English, the same word can be a verb, a noun and an adjective depending on the usage context. It is why the lookup mechanism is adapted to the grammatical category. Each category matches different datatypes. The table 8.1 shows the mapping correspondences between the business term category and the KB datatype classes. In some cases, several datatypes are listed for one grammatical category. It means that, if no match is found in the first datatype, a new mapping attempt is done with the following datatype and so on. If no match is found, the word is annotated as a fallback with a higher-class hierarchy, depending on the grammatical category.

Verbs without a detachable prefix, simple terms, adjectives and adverbs can be matched with a simple lookup step. Verbs denoting a status (conjugated in the past particle form) are first matched in the category '*FunctionConcept*'. The corresponding status concept stored under the '*ResultStatus*' class is retrieved via the '*hasCorrespondingStatus*' relation which associates these two classes. The matching of compounds, NPs and verb frames is more complex.

The first compound lookup step consists in a lookup against the basic lookup key. If a compound is matched, the associated concept is assigned. If no match is found, following steps are performed to match a potential synonym concept:

1. Each compound part is looked up in the KB and associated to one or more concepts.

Business Term Category	Datatypes	Fallback Datatype
NOUN	AbstractObjectConcept	AbstractObjectConcept
VERB + INFINITIVE	FunctionConcept	FunctionConcept
VERB + CONJUGATED	FunctionConcept	FunctionConcept
VERB + PAST_PARTICIPLE	ResultConcept	ResultConcept
ADJECTIVE	ColorConcept BooleanConcept ModiferConcept	AbstractStatusConcept
ADVERB	ColorConcept BooleanConcept ModiferConcept	AbstractStatusConcept

Table 8.1: Mapping table for grammatical categories

2. The different retrieved concepts are associated in all possible combinations in the same order as the compound parts, separated by an underscore.
3. The so-formed concept names are looked up in the KB. If a match is found, the concept is assigned to the compound. If not, the compound is associated to the higher class '*AbstractBusinessObject*'.

Example: the compound 'Abänderungsart'.

1. The lookup key 'ab_$ändern#art' is searched in the KB. No match is found.
2. 'Abänderung' is mapped to the concept 'CHANGE'
3. 'Art' is mapped to the concepts 'ART' and 'TYPE'
4. All concepts are combined in the compound parts order. It results in the strings 'CHANGE_ART' and 'CHANGE_TYPE'.
5. The lookup step gives a match for the concept 'CHANGE_TYPE', where the preferred form is 'Änderungsart'. The compound 'Abänderungsart' can be stored as a synonym of the 'CHANGE_TYPE' concept to avoid these processing steps next time when this term occurs.

The lookup of NPs works in an analog way. The considered NPs have the structure noun, followed by an article, followed by a noun, like 'Art der Änderung'. NPs comprising adjectives (like 'allgemeine Geschäftsbedingungen') are not treated as an NP because adjectives are considered as a noun modifier (see section 8.3.2). If an NP is stored in the KB, the basic lookup key is formed of the basic lookup key of the second noun, followed by a '#', followed by the basic lookup key of the first noun. The variant lookup key is set in the right order, to differentiate the NPs from their equivalent compounds. The first lookup step consists in a lookup against the basic lookup key in order to directly assign the right concept. If an equivalent compound is matched in this manner, the right concept is assigned as well. If the NP is not part of the KB, following steps are performed to match a potential concept:

1. The first noun is looked up in the KB and associated to one or more concepts.

2. The second noun is looked up in the KB and associated to one or more concepts.

3. The retrieved concepts are associated in the reverse order, separated by an underscore.

4. The so-formed concept name is looked up in the KB. If available, the NP is mapped to this concept. If not, the NP is associated to the higher class '*AbstractBusinessObject*'.

Example: the NP 'Art der Abänderung'.

1. The lookup key 'ab_$ändern#art' is searched in the KB. No match is found.

2. 'Art' is mapped to the concepts 'ART' and 'TYPE'

3. 'Abänderung' is mapped to the concept 'CHANGE'

4. The concepts associated in reverse order results in the strings 'CHANGE_ART' and 'CHANGE_TYPE'.

5. The lookup step gives a match for the concept 'CHANGE_TYPE', which has the preferred form 'Änderungsart'. The NP 'Art der Abänderung' can be stored as a synonym of the 'CHANGE_TYPE' concept to avoid these processing steps next time when this term occurs.

A fair amount of business verbs have separable prefixes. When used as a status in an event or in the infinitive form in process steps, the verb keeps its prefix like in following examples:

- Betriebsversammlungstermin festlegen
- Betriebsversammlungstermin festgelegt

However, if the verb is used in the active and present form (must be avoided according to the QA-model) the prefixes are detached like in the following example:

- Betriebsrat legt Betriebsversammlungstermin fest

If the attribute 'verbForm' is set to conjugated for a given verb, it has a potential detached prefix. The lookup step must retrieve all verbs which have the same stem in the KB (in the previous example 'legen'). In the handled business expression, the concept annotator checks the words that follow the verb. If one of the words corresponds to one of the detachable prefixes, the two verb parts are annotated with the associated concept. Otherwise, it is assumed that the occurring verb has no prefix. If a prefix is available, but was not recognized (not part of the KB), the business expression will normally not pass the structure annotator (next section).

Besides the concept assignment, the concept annotator marks simultaneously potential term usage errors. The usage form of the matched individual is checked during the matching process. The considered term is supplemented with a term usage error code. The correct form is retrieved via a KM lookup like explained in the 7.5. At this stage, four types of terminology errors are annotated:

- Deprecated term usage when the matched term individual has the '*UsageForm*' attribute set to '*deprecatedForm*' (annotation code 'TERM_DEPR').
- Spelling variant from preferred form usage, like hyphenation differences, equivalent NP instead of compound or grammatical variant. In this case, the preferred form was matched, but the word spelling is different (annotation code 'TERM_VARIANT').
- Abbreviation usage instead of long form, if the abbreviated form is a deprecated form (annotation code 'TERM_ABBR').
- Compounds or NPs having some synonym parts, like 'Auditprüfung'. The concepts 'AUDIT' and 'CHECK' are context-specific synonyms so that the term expresses an analog meaning twice. Such terms are annotated with the code 'TERM_DOUBLE'.

Example: the business expression 'Kundenangebot checken' results in following associations:

Kundenangebot	-> `BusinessObject`	–> CUSTOMER_OFFER
checken	-> `BusinessActivity`	–> CHECK –> TERM_DEPR

At this stage, ambiguities remain: if the LE analyses a word which belongs to different categories and cannot disambiguate them, several kind of concepts may be associated to the given term (for example business object and business activity). The structure annotator (see section 8.3.2) will help to disambiguate the business expression. Context-specific synonyms and relations are controlled and refined in the refinement annotator (see section 8.3.3).

8.3.2 The Structure Annotator

The analysis of the business repositories showed that the natural language expressions (erroneous or correct ones) assigned to the business objects mostly have similar structures. The idea is to define an extensible set of patterns recognizing the structure of business expressions. The patterns developed at this stage are deduced from the corpus. The patterns assign the concepts to information units, like object, subject, activity, status, modifier. While the KM defines only allowed patterns, the structure annotator recognizes also invalid patterns and provides a fallback mechanism that invites to reformulate if no pattern matched. The patterns include the usage of articles, auxiliary verbs or similar grammatical constructs that are not allowed by the guidelines. Business expressions are consequently marked with a specific compliance rule code.

The following examples are original business expressions extracted from the demonstration database delivered with the ARIS software. The table 8.2 shows examples of typical expressions for function and events objects. The third column details the different elements that compose the business expression.

Object Type	**Business Expression**	**Structure Elements**
Function	Rechnung drucken	*Object*: Rechnung *Activity*: drucken
Function	Finanzierungsunterlagen an UMG Bank senden	*Object*: Finanzierungsunterlagen *Activity*: senden *Destination*: UMG Bank
Function	Kunde begrüßen und ansprechen	*Object*: Kunde *Activity*: begrüßen *Second activity*: ansprechen *Activity coordination*: und
Function	Vermittler-Nr und Kennwort eingeben	*Object*: Vermittler-Nr *Second Object*: Kennwort *Object coordination*: und *Activity*: eingeben
Event	Barzahlung wird nicht gewünscht	*Object*: Barzahlung *Activity* : gewünscht Note: this expression is negated.
Event	Kunde hat Empfang quittiert	*Object*: Empfang *Activity* : quittiert *Subject* : Kunde
Event	keine Änderung notwendig	*Object*: Änderung *Status* : notwendig Note: this expression is negated.
Event	Unterlagen sind richtig/vollständig	*Object*: Unterlagen *Status* : richtig *Second status* : vollständig *Status coordination* : /
Event	Finanzierungsunterlagen sind nicht eingetroffen (>2 Wo)	*Object*: Finanzierungsunterlagen *Activity* : eingetroffen *Unit*: Wo *Count*: 2 *Comparator*: > Note: this expression is negated.

Object Type	**Business Expression**	**Structure Elements**
		Note: this is a parenthetical expression.
Event	Meldung 'Suche beendet' angezeigt	*Object*: Meldung *Activity* : angezeigt *Literal* : 'Suche beendet'
Event	Beschwerde automatisch eingereicht	*Object*: Beschwerde *Activity* : eingereicht *Activity modifier* : automatisch
Event	Vorbeugende Maßnahmen i.O.	*Object*: Maßnahmen *Object modifier* : Vorbeugende *Status* : i.O.
Event	Fahrzeug auf Lager	*Object*: Fahrzeug *Location* : Lager

Table 8.2: Examples of business expression structures

The elements which are displayed in the UML class diagram 8.3 are deduced from these typical examples. These elements are from now on called 'business expression part'. Some of the business expressions are negated, like 'Keine Änderung notwendig'. A Boolean attribute called '*negated*' is associated to the business expression to mark the negation.

In the examples, the strings associated to the business expressions parts remained in their original inflected form regardless of the previous concept annotator step in order to better highlight the different existing business expressions parts. The structure annotator consists of a set of pattern-based rules, called 'business expression templates', invoked after the homogenization and concept annotators. As described in the previous annotator sections, the terms used in the business expressions parts are annotated with their stem form, additional grammatical and semantic information and are mapped to their concept(s) or superclass(es). The cited information is exploited in the structure annotator.

The figure 8.4 depicts the metamodel used for describing the rules of the structure annotator. The central class 'BusinessExpressionTemplate' comes with a string attribute 'name', which should be a representative template name that allows a rapid visual identifying of the pattern structure. In this concept, the template names are based on examples, like 'kunde_hat_(nicht)_bestellt' or '(nicht)_konfiguriert'. The class 'BusinessElementTemplateSet' with its set of patterns is the container element for business expression templates. Patterns must be adapted for the different object types. The patterns described in the current section belong to the template set specific to the event object type ('TYPE_EVENT').

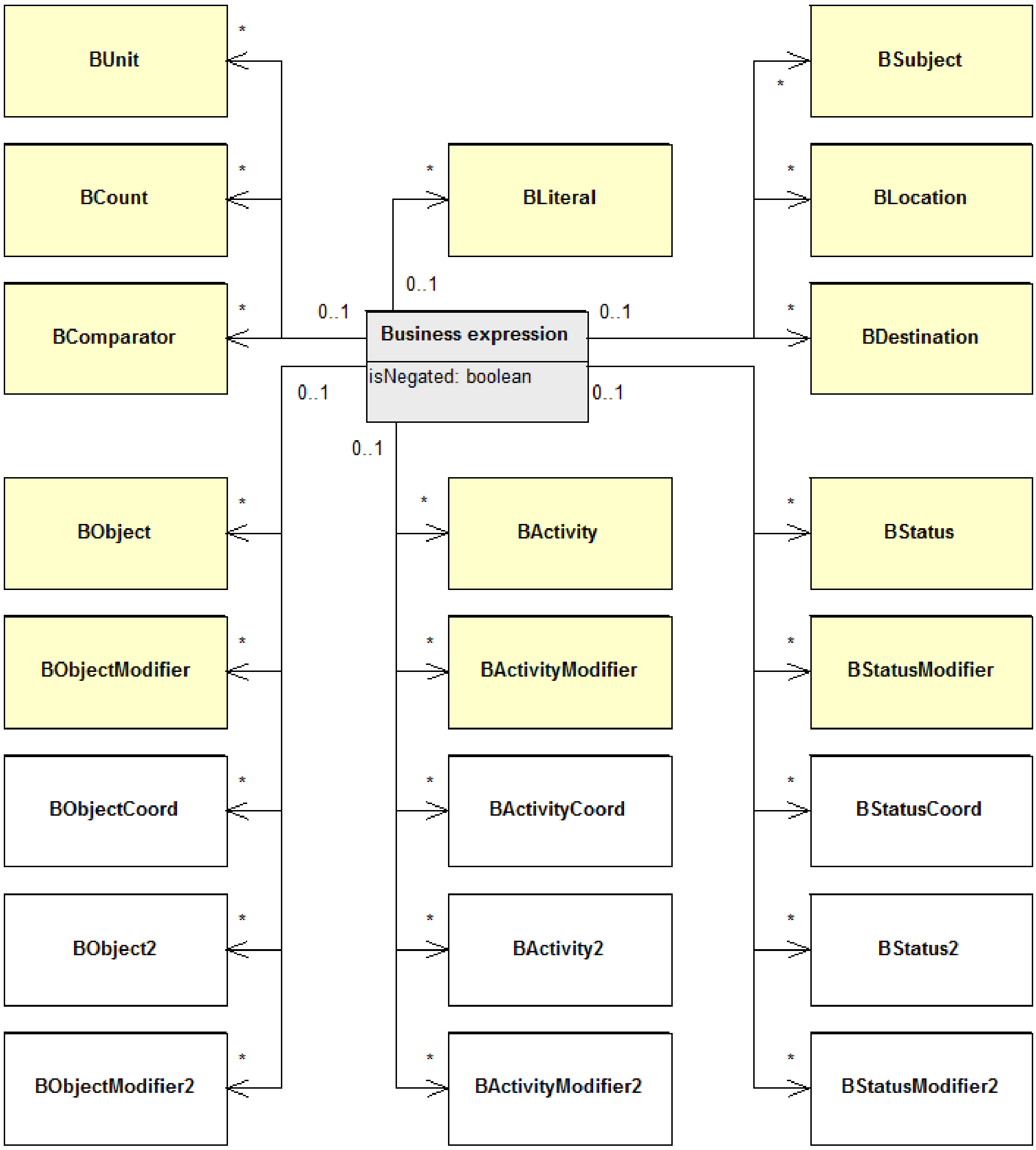


Figure 8.3: Elements of a business expression

The business expression template is composed of a sequence (at least one) of type 'BusinessExpressionTemplateElement', which consists of a set of condition attributes describing the pattern. The class 'BusinessExpressionTemplateElement' contains following attributes:

- *category*, string: used to match with the homogenized category name.
- *verbForm*, string: used to match with the homogenized verb form.

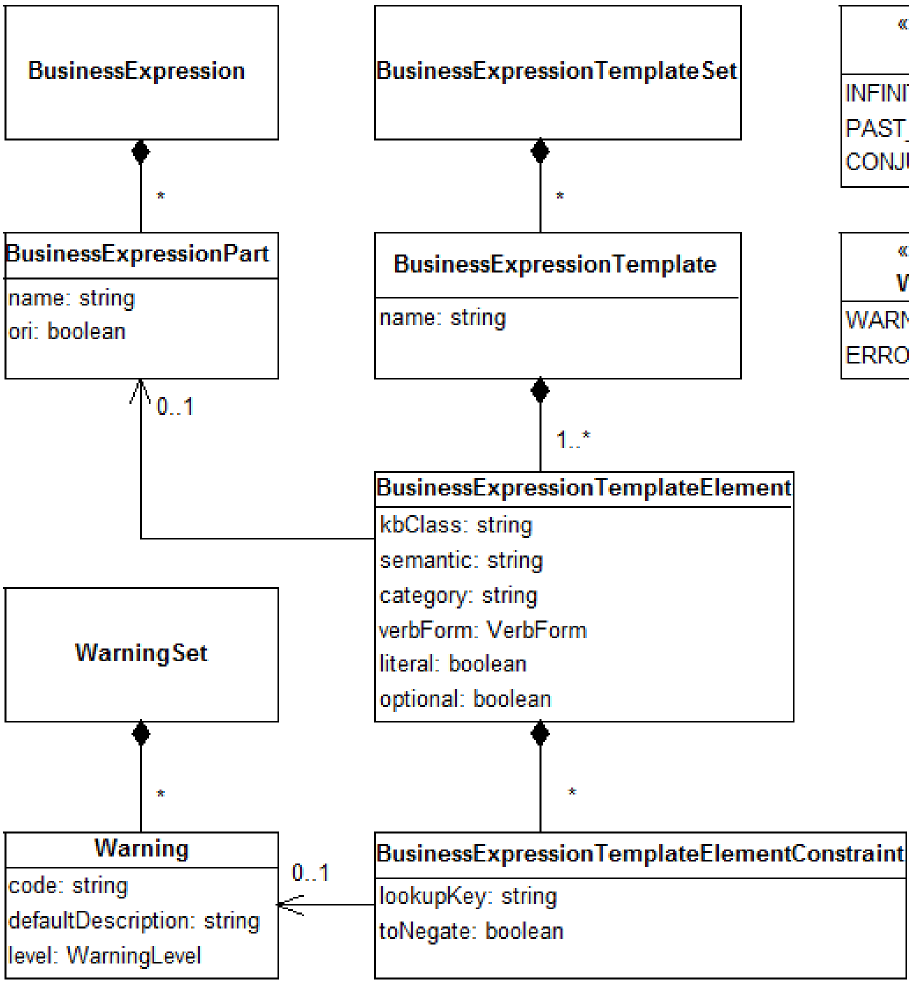


Figure 8.4: Metamodel of the structure annotator

- *semantic*, string: used to match with the semantic feature originally issued from the LE.
- *kbClass*, string: used to match a class structure in the KB. Can be a specific concept name class ('TO_SEND') or a superclass, regardless of the hierarchy level ('*AbstractBusinessConcept*', '*ObjectConcept*'...).
- *literal*, boolean: if true, the template matches a word sequence within simple or double quotes without interpreting the content.
- *öptional*, boolean: if true, the described element is only an optional part of the business expression and is not necessarily a matching element for the template.

If the 'literal' attribute is set to true, the other attributes are not interpreted because they can only match with a single word and not with a word sequence. In the main cases, the '*kbClass*' attribute is used instead of the '*category*' because the concept annotator should have mapped most of the terms to a KB class according to their category. Only some special cases like coordinations or punctuation marks are not settled by the previous step. The 'BusinessExpressionTemplateElement' class is used to describe patterns which do not need a reference to a specific word. For example, the business expressions 'Antrag vorhanden' and 'Rechnung korrekt', which have the same grammatical structure, are described with the following pattern:

BusinessExpressionTemplate.name: `antrag_vorhanden`

1. BusinessExpressionTemplateElement:
   kbClass: `AbstractObjectConcept`
   category:
   semantic:
   verbForm:
   optional: `false`
   literal: `false`
2. BusinessExpressionTemplateElement:
   kbClass: `AbstractStatusConcept`
   category:
   semantic:
   verbForm:
   optional: `false`
   literal: `false`

The template name 'antrag_vorhanden' does not denote a pattern searching for the words 'antrag' and 'vorhanden', but rather business expressions which have the same grammatical construct as 'Antrag vorhanden' (a noun followed by an adverb). In both template elements, the '*category*', '*semantic*' and '*verbForm*' attributes are empty. The empty strings are the default values for these attributes. In further examples, they will be omitted if empty. The same counts for the boolean values '*optional*' and '*literal*', which have the default value 'false'. The business expression matches if the concept annotator mapped the first involved term to one of the subclasses of '*AbstractBusinessObject*' and the second to one of the subclasses of '*AbstractStatusConcept*'.

Sometimes it is necessary to match a particular word. A good example is the negation recognition, which can not be detected only with the help of a grammatical category. In the German language, the negation is characterized by the words 'kein' or 'nicht'. For such purposes, an extension mechanism is provided, allowing the refinement of the general conditions expressed by the '*BusinessExpressionTemplateElement*' class. This is realized through the class '*BusinessExpressionTemplateElementConstraint*', which contains two attributes. A string '*lookupKey*' can be employed to match the '*lookupKey*' attribute associated to the handled term. A boolean 'toNegate' permits to declare the whole business expression as a negated expression if the constraint matches. Moreover, this class contains an association relation to the class *'Warning*', allowing the assignment of a warning code if the constraint specification matches.

The example below shows how different attributes are used in conjunction with constraints. As a naming convention for the template name, the optional parts are placed between parentheses. The *'optional'* attribute avoids a redundant definition of rules describing all the different possibilities that can occur. For instance, the next example stands for expressions like 'Kein neuer Antrag angelegt', 'Antrag ist nicht angelegt', 'graues Formular geprüft', 'Rechnung wird gedruckt', etc...

BusinessExpressionTemplate.name:
`(kein)_(neuer)_Antrag_(ist|wird)_(nicht)_angelegt`

1. BusinessExpressionTemplateElement:
   category: `OTHER`
   optional: `true`
   1. BusinessExpressionTemplateElementConstraint:
      lookupKey: `kein`
      toNegate: `true`
      warning: `STRUCT_EVT_KEIN`
2. BusinessExpressionTemplateElement:
   kbClass: `AbstractBusinessStatus`
   optional: `true`
3. BusinessExpressionTemplateElement:
   kbClass: `AbstractBusinessObject`
4. BusinessExpressionTemplateElement:
   kbClass: `BusinessFunction`
   optional: `true`
   1. BusinessExpressionTemplateElementConstraint:
      lookupKey: `sein`
      warning: `STRUCT_EVT_SEIN`
   2. BusinessExpressionTemplateElementConstraint:
      lookupKey: `werden`
      warning: `STRUCT_EVT_WERDEN`
5. BusinessExpressionTemplateElement:
   category: `OTHER`
   optional: `true`
   1. BusinessExpressionTemplateElementConstraint:
      lookupKey: `nicht`
      toNegate: `true`
6. BusinessExpressionTemplateElement:
   kbClass: `AbstractBusinessStatus`

In the previous example, the BusinessExpressionTemplateElement '1' describes a word which has the category 'OTHER' and is optional. As a constraint, it is specified that this word has a lookup key 'kein' – the stem form provided originally by the LE – that allows the matches of all inflected forms like 'keine', keines', keiner'... If this constraint matches, the whole business expression is a negated expression. A warning is then generated because in the German language, negated business expressions have to be formulated with 'nicht' instead of 'kein'. The BusinessExpressionTemplateElement '4' contains two constraints, each one fires a different warning.

The class '*WarningSet*' is a container for warnings represented by the class '*Warning*' that are specific to business object types. The class '*Warning*' has three attributes: a string '*code*', which is a short key to identify the warning, a string '*defaultDescription*' and a value '*level*', enumerated by the enumeration class '*WarningLevel*'. The level '*WARNING*' is used to mark formulations of business expressions which could be used unchanged depending on the usage context (controlled or restricted modeling). The level '*ERROR*' is used to mark business expressions which need an unconditional reformulation.

Guidelines are derived from the business repository analysis and the structure annotator results. The table 8.3 lists an extract of guidelines derived for event objects. The chapter 10 will give more insights into the quality assurance model for events as well as other object types.

Code	Level	Default Description
STRUCT_EVT_KEIN	ERROR	Use 'nicht' instead of 'kein'.
STRUCT_EVT_SEIN	WARNING	Do not use the verb 'sein' in events.
STRUCT_EVT_WERDEN	ERROR	Do not use verbs in the future or passive form.
STRUCT_EVT_ZU	ERROR	Do not use events containing 'zu'. Events must be expressed in the past form.
STRUCT_EVT_SOLLEN	ERROR	Do not use the verb 'sollen'. Events must be expressed in the past form.
STRUCT_EVT_MUESSEN	ERROR	Do not use the verb 'müssen'. Events must be expressed in the past form.
STRUCT_SLASH	ERROR	Events containing '/' must be transformed in two distinct events connected with an AND or OR operator.
STRUCT_AND	ERROR	Events containing 'und' must be transformed in two distinct events connected with an AND operator.
STRUCT_OR	ERROR	Events containing 'oder' must be transformed in two distinct events connected with an XOR or OR operator.

Table 8.3: Guidelines extract for event objects

From the previous examples, we can see that a side effect of the structure annotator is to allow warning annotations if the naming guidelines are not respected because the erroneous structures are recognized in the same step as the correct ones. Once the grammatical structure is determined, the structure elements must be associated to the classes defined on the UML class diagram 8.3.

The figure 8.4 shows an association between the class '*BusinessExpressionTemplateElement*' and '*BusinessExpressionPart*'. The class '*BusinessExpressionPart*' is the superclass of the elements depicted in the figure 8.3. It contains an attribute '*name*' which contains the reference to the element type (e.g. 'BActivity', 'BObject', 'BSubject'...) and a boolean '*ori*', which means that this element should keep its original content and will not be further processed in next steps. This is the case for literal elements, which consist in a quoted string that needs no other treatment. These '*BusinessExpressionParts*' are related to the class '*BusinessExpression*' because a business expression is composed of a sequence of business expression elements.

The relation between the class '*BusinessExpressionTemplateElement*' and '*BusinessExpressionPart*' has a cardinality of '0..1' because not all recognized structure elements are really business expression elements. For example, the verb 'sein' brings no valuable content to a business expression. The next examples highlight the association between the template elements and the business expression parts.

Example 1:

BusinessExpressionTemplate.name:
`(kein)_(rotes)_Auto_(ist)_(nicht)_auf_Lager`

1. BusinessExpressionTemplateElement:
   category: `OTHER`
   optional: `true`
   1. BusinessExpressionTemplateElementConstraint:
      lookupKey: `kein`
      toNegate: `true`
      warning: `STRUCT_EVT_KEIN`
2. BusinessExpressionTemplateElement: **–> BObjectModifier**
   kbClass: `AbstractBusinessStatus`
   optional: `true`
3. BusinessExpressionTemplateElement: **–> BObject**
   kbClass: `AbstractBusinessObject`
4. BusinessExpressionTemplateElement:
   category: `VERB`
   optional: `true`
   1. BusinessExpressionTemplateElementConstraint:
      lookupKey: `sein`
      warning: `STRUCT_EVT_SEIN`
5. BusinessExpressionTemplateElement:
   category: `OTHER`
   optional: `true`
   1. BusinessExpressionTemplateElementConstraint:
      lookupKey: `nicht`
      toNegate: `true`

6. BusinessExpressionTemplateElement:
   category: `OTHER`
   semantic: `loc`
7. BusinessExpressionTemplateElement: −> **BLocation**
   kbClass: `BusinessLocation`

The business expression 'neues Fahrzeug auf Lager' matches this template and results in following associations:

Fahrzeug	->	`BObject`
neu	->	`BObjectModifier`
Lager	->	`BLocation`

The correct business expression denomination can be found (if available) under the KM class '*BObjectModifierBObjectBLocation*'.

Example 2:

BusinessExpressionTemplate.name:
`(meldung)_literal_(ist|wird)_(nicht)_angezeigt`

1. BusinessExpressionTemplateElement: −> **BObject**
   kbClass: `AbstractBusinessObject`
   optional: `true`
2. BusinessExpressionTemplateElement: −> **BLiteral**
   literal: `true`
3. BusinessExpressionTemplateElement:
   category: `VERB`
   optional: `true`
   1. BusinessExpressionTemplateElementConstraint:
      lookupKey: `sein`
      warning: `STRUCT_EVT_SEIN`
   1. BusinessExpressionTemplateElementConstraint:
      lookupKey: `werden`
      warning: `STRUCT_EVT_WERDEN`
4. BusinessExpressionTemplateElement:
   category: `OTHER`
   optional: `true`
   1. BusinessExpressionTemplateElementConstraint:
      lookupKey: `nicht`
      toNegate: `true`
5. BusinessExpressionTemplateElement: −> **BActivity**
   kbClass: `FunctionConcept`

The business expression 'Meldung "Suche beendet" quittiert' matches this template and results in following associations:

Meldung -> `BObject`
quittieren -> `BActivity`
Suche beendet-> `BLiteral`

Example 3:

BusinessExpressionTemplate.name:
`(neue)_Unterlagen_(sind)_(nicht)_richtig_und|oder|slash_vollstaendig`

1. BusinessExpressionTemplateElement: **−> BObjectModifier**
   kbClass: `AbstractBusinessStatus`
   optional: `true`
2. BusinessExpressionTemplateElement: **−> BObject**
   kbClass: `AbstractBusinessObject`
3. BusinessExpressionTemplateElement:
   category: `VERB`
   optional: `true`
   1. BusinessExpressionTemplateElementConstraint:
      lookupKey: `sein`
      warning: `STRUCT_EVT_SEIN`
4. BusinessExpressionTemplateElement:
   category: `OTHER`
   optional: `true`
   1. BusinessExpressionTemplateElementConstraint:
      lookupKey: `nicht`
      toNegate: `true`
5. BusinessExpressionTemplateElement: **−> BStatus**
   kbClass: `AbstractBusinessStatus`
6. BusinessExpressionTemplateElement: **−> BStatusCoord**
   category: `COORD`
   optional: `true`
   1. BusinessExpressionTemplateElementConstraint:
      lookupKey: `und`
      warning: `STRUCT_AND`
   2. BusinessExpressionTemplateElementConstraint:
      lookupKey: `oder`
      warning: `STRUCT_OR`
   3. BusinessExpressionTemplateElementConstraint:
      lookupKey: `/`
      warning: `STRUCT_SLASH`
7. BusinessExpressionTemplateElement: **−> BStatus2**
   kbClass: `AbstractBusinessStatus`

The business expression 'Unterlagen sind nicht richtig/vollständig' matches this template and results in following associations:

Unterlage	->	`BObject`
richtig	->	`BStatus`
vollständig	->	`BStatus2`
/	->	`BStatusCoord`

If the structure is a correct structure (corresponds to a class in the KM), a new lookup can be done if all terms were previously mapped to KB concepts. The right formulation of business expressions could be found and a correction proposed. It is noticeable that the resulting structure detected by this annotator is not always a correct structure. Elements like '*BStatus2*' or '*BStatusCoord*' are only available in order to recognize an unwished structure while the KM contains only correct structures. These helper classes can be extended, if needed, to enlarge the spectrum of matching business expressions. The correct elements are highlighted in yellow in the UML diagram '8.3'. The last purpose of the structure annotator consists in a grammatical disambiguation. Sometimes the LE analyses a term with several categories. The rules are ordered from the simpler and more general cases to the more specific ones. As soon as a template matches for one of the analyzed categories, it is assumed to be the right one and the others are eliminated. The appendix A lists an extract of implemented templates as well as one or two matching business expression examples for each rule.

Technically, the structure annotator is realized as an EMF application. The UML class diagram depicted in figure 8.4 was automatically generated from an ecore metamodel which was created with the EMF plugin for the eclipse development environment. Once the metamodel is defined, corresponding models describing the business expression templates are created (listed in appendix A). In the figure 8.5, these models are called 'EMF templates' and are input of the openArchitectureWare component, as well as some rule files (in the figure 'Rules.xpt') which describe how the output of openArchitectureWare looks like.

In this case, the output consists of java files containing procedures for all rules defined in the models in a declarative and sequential manner. The order of the rule declaration is thus important because the annotator stops the processing as soon as a rule matched. This strategy has the advantage of performance because no rule interpreter is needed and all rules are hard-coded after the generation process. The hard-coded rules bring no disadvantages because they are defined in a flexible manner by the model. The generated java code is integrated in the whole prototypal project.

In this annotator step, the structure of a business expression is interpreted and valuable structural formulation errors are identified. Some business expressions are not recognized because their structures do not match any of the templates. In these exotic cases, the user should be requested to reformulate because a balance must be found between template definitions and the number of potential rule matches.

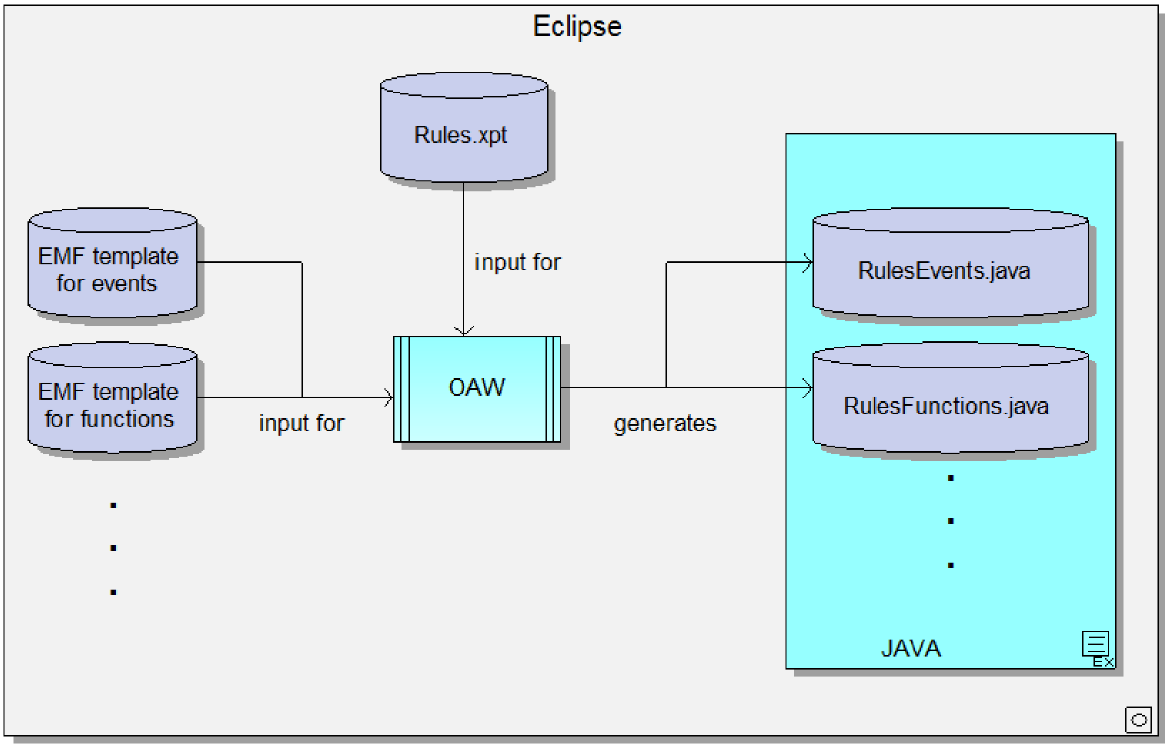


Figure 8.5: Rules generation process

8.3.3 The Refinement Annotator

At this stage, terms are mapped to concepts and the business expression parts are recognized. The concept annotator could not fulfill all requirements due to missing structure information, which is now available. The refinement annotator performs several tasks on conceptual level. If the previous annotator step failed, this annotator cannot further refine the results.

The first task consists in a contextual disambiguation. While the structure annotator could disambiguate on grammatical structure level, some terms (homonyms) can still be associated to several concepts, like for example 'warten', which is associated to the concepts 'TO_MAINTAIN' and 'TO_WAIT'. In such cases, context information is taken into account, based on the dynamic relations defined in the KB (see table 7.10). The structure annotator recognized the different business expression parts. This information serves to lookup the usage restrictions between the parts. For example, the term 'Datenbanksystem' is associated to the concept 'TO_MAINTAIN' but not to 'TO_WAIT'.

Once the disambiguation step was performed (if possible!), the restriction usages are further checked. With help of the dynamic relation definitions, the relations between business objects, activities, statuses, object modifiers and activity modifiers are controlled. Three cases can occur:

- The dynamic relation between the handled business expression parts is allowed. No additional annotation is generated.
- The dynamic relation between the handled business expression parts is forbidden. An error annotation is produced (code 'TERM_RELATION') notifying the usage constraint violation.
- The dynamic relation between the handled business expression parts is undefined. An annotation is produced (code 'GOV_UNDEFINED_RELATION'), which will trigger an input request for the KB via the business collaboration platform.

In case of forbidden or undefined relations, the annotator checks for an appropriate alternative via context specific synonyms. For example, the business expression 'Produktionsqualität prüfen' results in following associations:

Produktionsqualität	->	`BObject`	-> PRODUCTION_QUALITY
prüfen	->	`BActivity`	-> TO_CHECK

The term 'prüfen' is the preferred form of the concept 'TO_CHECK' so that normally no error would be fired. However, no dynamic relation is defined between the concepts 'TO_CHECK' and 'QUALITY' while a relation '*isAssociatedTo*' is set between the concepts 'TO_CONTROL' and 'QUALITY'. Relations concerning the unspecific term 'QUALITY' are per default inherited by its subtypes. As the concepts 'TO_CHECK' and 'TO_CONTROL' are context-specific synonyms, the refinement annotator proposes the usage of 'kontrollieren' instead of 'prüfen'.

A very specific task concerns the reformulation of business expressions using the verb 'durchführen'. For example, the business expression 'Benutzeranmeldung durchführen' involves the business object 'Benutzeranmeldung', which is stored in the KB under the category '*ActionConcept*'. The compound head 'Anmeldung' has a static relation '*hasCorrespondingFunction*' to the verb 'anmelden'. The refinement annotator proposes the expression 'Benutzer anmelden' instead. Note that not all expressions can be reformulated in that way. For example, 'Anmeldung durchführen' would result in 'anmelden' without any business object. In this case, business objects related to the activity 'anmelden' can be proposed.

As a last step, the refinement annotator marks the unspecific terms with the error code 'TERM_UNSPECIFIC' after a lookup in the KM to determine whether the term is really unspecific or not (negative or positive data hierarchy) like explained in the section 7.5.1.

8.3.4 The Spelling Annotator

All terms not recognized by the LE are marked by a flag '*unknown*' set to true in the homogenization annotator. The concept annotator may have mapped an unknown word to a business concept. In this case, this term is supposed to be a regular business term. The spelling annotator then resets the '*unknown*' flag to false. If a term remains unknown after the concept annotator step, the spelling annotator performs a fuzzy match like standard spelling checkers to propose a correction. The difference is that the KB is used as the authoritative resource to propose business specific terms.

8.3.5 The Grammar Annotator

An optional grammar error annotator could be added at this stage. On one hand, the set of needed rules is very restricted for German because business expressions are no regular sentences. The only really interesting rule would be the congruence between adjectives and nouns. The correct verb forms have already been checked by the structure annotator. On the other hand, only a minor count of such errors were found while analyzing the corpus. Thus, this is not the point of interest in this dissertation.

8.3.6 The Interface Annotator

The interface annotator 'localizes' business expressions in a technical language used as a programming blueprint which favors a rapid transformation into an IT solution. The resulting names can be considered as an Application Programming Interface (API) that represents the technical implementation interface of a process step that can be deployed as a service in a Service Oriented Architecture.

The KM entries are supplemented with the attributes '*technicalName*' and '*technicalPluralName*'. If the business expression has already been stored in the KM, the interface annotator only looks up the corresponding name. Otherwise, the complete technical name must be built from the technical names associated to the concepts involved in the business expression. The concatenation follows specific rules adapted for each correct business expression structure. If the structure is incorrect, no API name is generated. Following rules apply:

- The API name starts with a lower case letter.
- The API name stands as a concatenation of the technical names of the concepts involved in the business expression in a given order, which is specific to the business expression structure.
- Each concatenated string starts with a capital letter.

Following examples correspond to the SOA example depicted in the section 5.4.5:

- Event 'Aktienverkaufspreis festgelegt': 'shareSellingPriceDetermined'
- Function 'Aktiendaten abfragen': 'queryShareData'
- Event 'Aktien verkauft': 'sharesSold'

After the SE-BPM processing chain has been completed, the API name should be stored as an attribute in the business repository for each concerned business object. Model transformation procedures like the transformation of an EPC into a BPMN or a BPEL formalism must be adapted to read this attribute and use it as the transformed object name.

If the business object ('*BObject*') occurring in the business expression is in the plural form – attribute '*plural*' of the homogenization annotator set to true – the business expression is considered to be in plural. The same concatenation rules apply, but the attributes '*technicalPluralName*' are used for nouns instead of '*technicalName*'.

In a business-driven SOA, technical diagrams like BPMN or BPEL are exported from the modeling environment into a developing environment to supplement it with technical attributes and bring the processes to execution. Sometimes, the developers are adding activities connected to software services (orchestration) in the implementation phase, without first updating the business model. If all implemented services in a SOA were originally derived from a business process step, the new activity name would consist in the interface name previously generated by the interface annotator. In a round-trip scenario, the diagrams are reimported. If the technical name remained standardized, a mapping can be done between a business process step name and the technical function name so that the business model can also be updated. The technical name is split in its base units (before concatenation) and mapped to concepts. From the concepts, the business expression name is retrieved from the KM or generated with the same technique as the KM entries (see 7.5.2).

8.3.7 The Governance Annotator

The governance annotator annotates business expressions with processing instructions to act as an input for the governance level. These annotations are not intended for the end user (business modeler), but rather for the business content expert. Following information is added in this step:

- All terms which were not mapped to a concept by the concept annotator are marked with the code 'GOV_NEW_TERM'. If the term is marked as unknown, the code is set to 'GOV_NEW_UNKNOWN_TERM' because it is not clear whether this term is a real spelling error or a regular business term, but unknown to the LE. If the term was recognized by the LE, the semantic value is interpreted to make a proposition for associating the term to a specific class of the KB, according to the methods described in the previous chapter ('*AgentConcept*', '*ActionConcept*'...).
- Like explained in the concept annotator, compound terms or NPs can be matched as potential synonyms even if they are not registered. These terms are annotated with a code 'GOV_VARIANT' in order to be validated by the business content expert so that the next time this term occurs, it will be mapped to a concept

with a simple lookup step and thus potential similar handling errors are avoided if the term mapping was erroneous.

- If the business expression structure was recognized and all concepts have already been registered, but the business expression is not yet part of the KM, it is annotated with the code 'GOV_KM_ENTRY'. The entry can then be created automatically with the '*entryStatus*' attribute set to '*generated*' for further validation purposes or it can first pass a validation step by the business content expert before storing. This depends on the choices met on governance level. KM entries generated in this way are considered to be a collaborative work because the KM entries are growing due to business modeler's input.
- If the business expression structure was not recognized, it is annotated with the code 'GOV_UNKNOWN_STRUCTURE'. If the considered structure has to be recognized as well, a new structure annotator template must be implemented.
- Missing relations between concepts have already been marked in the refinement annotator with the code 'GOV_UNDEFINED_RELATION'. The business content expert must determine whether this relation is always allowed, always denied and on which term hierarchy level (subtypes).

8.4 The SE-BPM Consumer Applications

The chain of producer annotators have marked the handled business expressions with enough information to allow different applications to consume the data in order to fulfill the requirements described in the use cases chapter. The consumer applications can either work on the internal object structure produced by the annotators or work on an standardized output format like RDF (see section 3.3.1). As a recommendation, the result should be stored in the business repository with help of different attributes associated to the handled business object in order to extend the existing business process repository with a machine understandable semantic layer. The consumer applications can then directly work on the produced semantic information.

8.4.1 The Quality Checkers

The most obvious application scenario concerns all use cases involving quality checkers: restricted modeling (see 6.2.1.1) and the quality check report (see 6.2.1.3). The controlled modeling use case does not belong to this category because all input created with help of the controlled modeling wizards are correct business expressions and missing business expressions must first be requested.

The quality checkers involve a spelling checker, a terminology checker and a compliance checker (and the optional grammar checker) to control the business expressions in terms of correctness, appropriateness and readiness. They interpret the previous annotators results and format the output in a user-friendly way, like transforming the annotation codes in user-understandable error messages and highlighting the concerned business expression parts. The quality check report displays a list of objects which do not respect the quality metrics defined by the quality assurance model (see chapter 10). In the restricted modeling use case, the error messages are displayed on the fly when creating the business objects in the modeling environment.

8.4.2 The Consistency Applications

Optimization tasks like business repository consolidation (see use case 6.2.2.1) and process merging (see use case 6.2.2.2) require a consistent business repository to complete as efficiently as possible. If the business repository was built with controlled modeling, it should already be consistent and the consolidation tasks consist then only in retrieving objects having the same name. However, if restricted modeling only was adopted, the quality checkers are basing only on standard business resources (standard KM and KB) and undefined relations or unregistered terms are only checked on a low level like general compound hyphenation rules. A certain amount of inconsistencies can remain in that case. Then again, while consolidation tasks are working on one local business repository, merging tasks can involve external business processes, which were never checked against the QA-model (for instance in case of company acquisition).

The business repository consolidation and process merging are working in the same manner, basing on a consistency checker, with only one difference: for consolidation, all objects of the business repository are compared to each other, for process merging, only the objects occurring in the handled business process are compared to the business repository content. The first consistency checker step compares the business objects using the semantic layer. Following procedure is executed:

1. Retrieval of business expressions mapped to the same structure and the same concepts independently of the object denomination. This step finds out all business expressions which have:

   - The same denomination (already consistent). In the consolidation use case, this step is not necessary because these business objects have already been consolidated. For process merging, this step is required because the objective is to find similar process steps.
   - The same base denomination, but firing some rule compliance errors: 'Bewerbungsunterlagen sind vorhanden' instead of 'Bewerbungsunterlagen vorhanden'.
   - The same denomination, but inconsistent usage of singular and plural: 'Reisebelege prüfen', 'Reisebeleg prüfen'.
   - Business expressions using synonym terms (variants of preferred forms or deprecated forms): 'Reisebelege checken' instead of 'Reisebelege prüfen'

2. Retrieval of business expressions expressed in a negative formulation corresponding to the positive formulation: 'Reisebelege ungültig' vs. 'Reisebelege nicht gültig'. This check is based on the '*hasAntonymBoolean*' relation associating two statuses in the KB.

3. Retrieval of business expressions mapped to the same structure, but where business terms have a different term specification level (unspecific terms): 'Belege prüfen' instead of 'Reisebelege prüfen'.

4. Retrieval of business expressions mapped to a larger or smaller structure (optional object modifiers, activity modifiers or status modifiers), but involving the same concepts: 'Reisebelege prüfen', 'neue Reisebelege manuell prüfen'.

5. Retrieval of business expressions mapped to a larger or smaller structure and having a different term specification level: 'Reisebelege prüfen', 'neue Belege manuell prüfen'.

6. Retrieval of business expressions mapped to the same structure, involving the same business objects, but related to context-specific synonyms: 'Reisebelege prüfen', 'Reisebelege kontrollieren'

7. Retrieval of business expressions mapped to the same structure, but with business terms having a different term specification level and being related to context-specific synonyms: 'Belege kontrollieren' instead of 'Reisebelege prüfen'.

8. Retrieval of business expressions mapped to a larger or smaller structure, involving the same business objects, but related to context-specific synonyms: 'Reisebelege prüfen', 'neue Reisebelege manuell kontrollieren'

9. Retrieval of business expressions mapped to a larger or smaller structure, with business terms having a different term specification level and being related to context-specific synonyms: 'neue Belege kontrollieren' instead of 'Reisebelege manuell prüfen'.

10. Mix of all the previous steps considering negative and positive formulations: 'neue Belege nicht gültig', 'Reisebelege ungültig'.

The previous steps are based on a conceptual level and the KB/KM serve as the reference resource. If the structure annotator matched, but no or not all concepts were associated to the terms, the previous steps are repeated with one difference: the stem forms of terms are examined instead of concepts. Following criteria are observed:

- Grammatical variants: 'Prüfkonzept', 'Prüfungskonzept'

- Hyphenation variants: 'Rohbau-Toleranz', 'Rohbautoleranz'

- Synonym variants: if a compound is used, some of the parts can be mapped to concepts. That can bring some synonym information: in a municipal administration, a process can be modeled describing the extermination of bedbugs. 'Bettmatratzenprüfung' and 'Bettmatratzencheck' are synonyms because 'prüfung' and 'check' are well-known concepts, while 'Bettmatrazen' is not part of the standard business vocabulary.

- Different specification levels: 'Matratzenprüfung', 'Bettmatratzenprüfung'.

In some cases, the concept annotator could not annotate all concepts and the structure annotator also failed. A structure annotator failure can happen in case of restricted modeling, but during modeling time the business modeler is invited to reformulate. Such a failure can then only happen in two cases: the business modeler did not reformulate and, in the process merge use case, the external processes were never checked against the quality assurance model. In these circumstances, only stem forms are matched and the found business expressions can be semantically distant.

All the cited steps are retrieving business expressions ordered from the most exact matches to the most distant ones. At governance level, options may be set to configure the wished matching categories.

8.4.3 Retrieval in the Business Repository

Process findability is one of the main requirement in BPM: during design time for impact analysis (see the use case 6.2.5.1), for supporting the modeling tasks and for favoring the process execution with the retrieval of published processes (see use case 6.2.4.1).

The retrieval functionalities are following the same algorithm as the consistency applications (see 8.4.2) with one major difference: the input. Consistency applications are always comparing two business expressions. The input of a search process can be very different: a simple business term, a business activity, a complete business expression or even two or more terms which are not related. The search string is sent to the LE annotator, the homogenization annotator and the concept annotator. If all terms are mapped to concepts, the search is performed on the semantic layer according to following algorithm:

- Retrieval of business expressions containing all concepts comprised in the search string.
- Retrieval of business expressions containing at least one of the concepts comprised in the search string (ordered by concepts count).
- Retrieval of business expressions containing a different specification level (more or less specific) than the search concepts.
- Retrieval of business expressions containing context-specific synonyms of the searched concept.

If no or not all search strings were mapped to concepts, the same algorithm is applied on the term stem. The stems are then compared to elements of the business repository that are not mapped to concepts. If the business repository was elaborated with controlled modeling, all business objects have already been conceptualized. In this case, the search string should be sent as a term request to the collaboration platform.

When processes are published in the intranet, the process retrieval can be an interaction between tags associated to the business process and search strings (see the use case 6.2.4.2). The tags can be proposed automatically once the semantic layer for each element of the business process is produced. The preferred tags are the involved business objects. The automatic tag proposition prevents from entering manually tag variants and thus remains KB-conform. When entering a search string, the processes marked by corresponding tags are at first retrieved before checking the content of each involved object. The results should be more accurate because the tags are explicitly selected.

8.4.4 Governance Applications

Governance applications are interpreting all entries marked by the annotation code starting with the prefix 'GOV_'. These entries are sent to the collaboration platform for further validation. The business content expert retains the authoritative role to decide whether the entries or the requested terms are correct or not. The associated use cases are the business term requests (see 6.2.6.6) and the business database mining (see 6.2.6.3). The business term requests are mainly issued from an interactive process coupled with the controlled modeling wizard or resulting from the annotator chain in a restricted modeling use case. The business database mining is a batch process during which all unchecked business objects are sent through the SE-BPM engine in the same way.

8.4.5 Localization Process

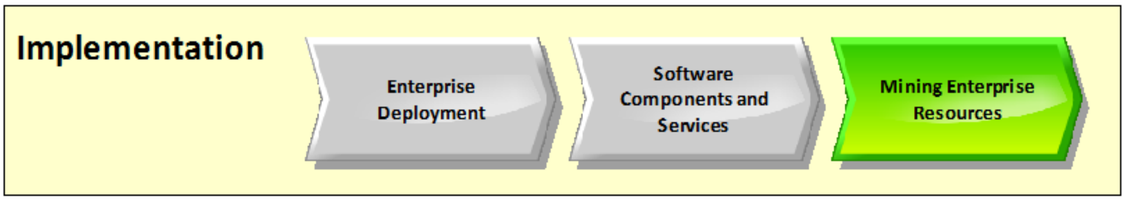
In a multi-language BPM environment, the KM is the main translation source for business expressions. If a given business expression concept entry has an equivalent in the target language, the language attribute of the business object stored in the business repository can be filled on the fly by the target language value. If the needed target value is not set in the KM, it can be generated like described in the section 7.5.2.

If a new language is introduced, localization software can be deployed to support the translation of the KB and the KM. This issue is out of scope of this dissertation, but the conclusion describes some translatability perspectives.

This chapter described the implementation and deployment of semantically-enriched BPM applications. The components and the according workflow are tailored to check the quality of business objects stored in business repositories. However, the BPM activities involve many other resources like process documentation, service descriptions, terminologies or structured data stored in data warehouses. The next chapter shows how to reuse and enhance the developed components for exploiting structured and unstructured enterprise resources.

Chapter 9

Mining Enterprise Resources



Enterprise structured and unstructured resources store valuable information, which can be reused in the Business Process Management life cycle. The quality management framework provides software components for gaining data from enterprise resources, integrating them on the governance level and exploiting them during the BMP life cycle. Following aspects are treated:

- Support on governance level: the business-specific knowledge models store business terms gathered while analyzing the corpus business repositories. Knowledge models are growing in a collaborative way, supported by term request functionalities during modeling time. However, terms needed at modeling time often are already part of legacy business repositories, corporate terminologies, existing domain-specific ontologies or occur in unstructured process documentation. Registering relevant business terms before the business modeler needs them reduces the modeling efforts.

- Support during modeling time: process documentation is often the input of modeling activities. Extracting information and generating business objects help reducing the modeling time. The change management process is supported by checking whether the process documentation is properly aligned to the corresponding business process model.

- Support during execution time: if a human process participant is not aware of some process steps, he may consult process models published in the company's intranet. Some complex process steps need to be further documented. Techniques which help linking process documentation to specific process steps are described.

- Support during implementation time: the method developed for interpreting process documentation is adapted to natural language service descriptions coming with Web services to favor service discovery in a SOA.

9.1 Mining Terms from Structured Resources

Companies are deploying several back-end systems storing useful enterprise data on different supports. Structured enterprise data can contain business terms relevant at modeling time like natural language terms or items which are not part of natural language like IT system names, organizational unit names or even personal names. The main characteristic of structured resources is that the stored elements already have an intrinsic semantics that facilitates the registering of the concerned terms.

9.1.1 Mining Terms from the Business Repository

Mining terms from the business repository supports the use case 'Business Database Mining' (see 6.2.6.3). The business repository is the central structured resource of the application scenario. The business repository is a particular case: it acts simultaneously as a source and a target information base for business vocabulary. Its input can serve to extend the KBs and in the same time, business models can be extended by information gathered from external resources like explained in the following sections. The data flow must be strictly defined on governance level.

Information issued from some specific business models like IT infrastructure or organizational charts is reused in process models. As a recommendation, all legacy business repository contents created prior to the introduction of controlled modeling should be exported and occurring business objects should be registered in the company-specific KB. Object types that have a clearly defined semantics like IT systems or organizational units can be automatically registered in the according KB class. The registering of the remaining business terms is supported by the techniques described in the section 7.3.2. Once the legacy repositories were mined, the registered terms can be reused in subsequent modeling steps. Organizational charts, IT infrastructure and IT landscape diagrams should as well belong to the model types that are modeled with a controlled modeling wizard. This makes sense because the containing objects can be inconsistent, for example 'MS Word', Microsoft Word', 'Word'...

Technical term models are a special case. In a controlled modeling environment, technical term models can only be created with the wizard and the needed vocabulary must first be requested. Once the model was created, it brings additional information which is not yet part of the KBs: semantic relations between technical terms. Besides grammatical relations, the only semantic relation available in the KB is the specification relation between compounds. For example, a 'Reisebeleg' is a 'Beleg'. But a 'PKW' – perhaps declared as a context-specific synonym – is not declared as a subtype of 'Fahrzeug' . The latter relation can be modeled in the technical term model. These relations can be exported in the company-specific KB and be exploited by the SE-BPM engine like in the 'Process Consolidation' use case (see 6.2.2.1).

9.1.2 Mining Enterprise Data Repositories

In the application scenario, the term 'enterprise data repository' is used as a generic term which encompasses all specific software which can store enterprise data in a structured way like databases (relational, object-oriented or XML), data warehouses or Lightweight Directory Access Protocol (LDAP) directories... Enterprise data repositories may store strongly typed data that is of interest for business modeling, like the previously cited personal names, IT systems or organizational units. A bridge can be built between the KB and the enterprise data repository in order to register automatically the needed or discovered terms. The difficulty is to define the data flow direction to avoid inconsistencies between the business repository, the KB and the enterprise data repositories. The data flow depends on how the data is filled on the enterprise data repository side. The question to answer is whether the modeling tasks serve as an assessment of the enterprise structure or provide requirements to extend the IT infrastructure or reorganize organization units. The data flow direction can be defined according to this strategy.

9.1.3 Mining Semi-Structured Documents

Semi-structured documents contain partly free-form text and partly markers to emphasize semantically typed units. The most common example of semi-structured documents are XML documents with tags as markers. Some of the tags may contain interesting information from the business modeling point of view. For instance, a printing machine manufacturer may model some printing processes including specific colors that are not part of common color names. If a process description document written in XML serves as the modeling input, a specific tag marking text portions as color name may be defined. The values can then be extracted and mapped to a special category in the KB. Of course, this also applies to standard business objects like IT systems or organizational unit names. In this use case, the data flow must also be defined in order to specify which is the authoritative source delivering the information marked by XML tags to avoid redundancies and inconsistencies.

9.1.4 Mining Corporate Terminological Resources

The previous sections concentrated on strongly typed data that are in the majority of cases not part of the natural language vocabulary. Companies often define their own domain-specific terminologies or ontologies for different purposes like the writing of user-guides, maintenance manuals or repairing instructions. Terminological resources may consist in domain-specific ontologies deployed on the internet, for instance medical lexicons or the definitions of country and city names. A part of the terms defined in the corporate terminological resources may be used in the business modeling context and thus be requested sooner or later by business modelers.

The use case 'Business Terms Mining from Terminologies' (see 6.2.6.5) describes the KB extension with terms extracted from corporate terminologies. The difficulty is inherent to the nature of corporate terminological resources: if the resources had grown historically, they may be inconsistent. Even if some companies are strictly administrating their terminologies so that the term entries are of highest quality, the difficulty resides in the amount of terms because not all terms are business-relevant. A compromise must be found to prevent from registering all terms and favoring the extraction of relevant ones.

Following procedure helps to filter out terms that can be relevant, basing on attributes defined on term entries level and on the procedure described in the section 7.2.4.4 for separating domain-specific terms from common ones:

1. All terminology entries (or the newest entries only if this procedure was already performed) are sent to the first annotator steps: LE annotator, homogenization annotator and concept annotator in order to conceptualize the terms and thus cope with potential inconsistencies, synonyms and subsequent registering activities (like described in 7.3.2).

2. Terminological entries often come with a marker denoting a specific domain like 'repair instructions'. If one of the markers corresponds to relevant business terms, the according terms are extracted.

3. For each compound or NP, the head is extracted and compared to the simple terms available in the standard KBs. If a match is found, the corresponding terms are proposed as subtypes of standard business vocabulary. Depending from which KB the simple term was extracted (standard or domain-specific), the extracted term is supposed to belong to the corresponding domain.

4. Terms related to terms occurring in the KBs are extracted: NPs or compound terms with only one different part, for example 'Nahverkehrstour' vs. 'Nahverkehrsroute'.

5. As an optional step, compound parts occurring in the KB (not only heads) are ordered according to the weight of the container concepts. Terms sharing the same compound parts are retrieved and ordered in the same way. This step can bring some irrelevant terms, which can be disabled by adding them to a 'stop-word' list (check box 'Ignore ...' in the figure 6.47).

6. If possible, a process map entry is proposed per default according to the classification of the term's supertype.

Extracted terms can be recognized as potential synonyms of KB terms. In the collaboration platform, the business content expert decides whether such terms are really synonyms or represent an own concept. The problems which occur in this situation are clashes between preferred/deprecated term definition in the terminological resources and in the KB. This problem occurs as well with terms stored in the standard KBs and which are also available in the terminological resources. Both resource types should be consistent. Two decisions can be taken: either the enterprise terminology is adapted to fit the standard business resources, or the standard (and domain-specific) business KBs are overwritten with entries in the company-specific KB according to the terminological resources. In the latter case, it is important to extract the terms from the terminological resources upon or even prior to the introduction of controlled modeling to avoid clashes in the business repository due to the usage of different KBs by the SE-BPM engine. If the SE-BPM engine does not rely on the standard business vocabulary, it limits the interoperability of business models to the enterprise. Further process merging activities can then be affected, if the companies or even departments do not share the same terms. In contrary, if domain-specific or general business terms which were not part of the domain-specific or standard KBs are extracted, the business content expert can choose to share them with the business modeling community and in this way, collaborate to create common and interoperable business terminological resources.

9.2 Mining Terms from Unstructured Resources

Unstructured resources handled in this section concern all documents containing free-form text, independently of the document format. Service descriptions are the first kind of treated unstructured documents and more complex process description documentation is the second kind. The handling is based on components of the SE-BPM engine to take advantage of the conceptualization to deal correctly with natural language term variants.

9.2.1 Software Service Annotations

Software service descriptions are formal documents describing the interfaces including operations, input and output parameters of a software service. Such a description often comes with a natural language part, highlighting the functionality of the service. Although service descriptions are structured documents, the natural language part remains unstructured. This part, containing valuable information about the service functional and non-functional capabilities is the point of interest in this section.

To extract the information contained in the description, the description is sent to the first annotator steps of the SE-BPM engine: LE annotator, homogenization annotator and concept annotator. A consumer component interprets the results of the cited annotator chain:

1. All concepts belonging to a subclass of '*AbstractObjectConcept*' are extracted (only nouns and NPs) to find out business objects affected by the service, called service descriptors. Terms which were not conceptualized are ignored.

2. The extracted concepts are associated to process map elements in order to classify the concepts in business aspects. This step will only be successful for specific terms.

3. Service descriptions probably contain a set of unspecific terms because the authors who write the descriptions know the context well and will not always choose very specific terms. According to the process map association performed in the previous step, specific terms deriving from the unspecific terms and related to the same business aspect are proposed.

4. As an optional step, depending on the available data, the relations created in technical term models can be interpreted to find semantic contiguousness between terms and so, map them to the same business aspect.

5. All concepts belonging to a subclass of '*ModiferConcept*' are extracted (adjectives and adverbs) to find out non-functional service capabilities. Not all modifiers are non-functional capabilities. To filter them out, the KB model requires a new boolean attribute, called '*soaCapability*', which is set to true if the considered adjective or adverbs can be a non-functional requirement.

The procedure is highlighted by a concrete example based on the use case 'Service Annotation' (see 6.2.3.1). The software service description handled by the use case is a WSDL file describing a Web Service which delivers stock exchange information. The figure 6.29 shows an import step of the WSDL file and displays highlighted business terms. Service descriptors are generated according to this description. For reminder, the example Web Service comes with following natural language description:

> *StockExchangeDAX liefert auf Anfrage echzeit Aktiendaten vom Frankfurter DAX. Dieser WebService liefert die Daten vom letzten Verkaufszeitpunkt, wie Preis, Volumen und Zeitpunkt.*

Following terms belonging to the subclass '*AbstractObjectConcept*' (first column) are conceptualized (second column) and are associated to a process map element (third column).

Aktiendaten	->	SHARE_DATA	->	Aktienhandel
DAX	->	DAX	->	Aktienhandel
Verkaufszeitpunkt	->	SELLING_TIME	->	Vertrieb
Zeitpunkt	->	TIME		
Preis	->	PRICE		
Volumen	->	VOLUME		
Anfrage	->	INQUIRY		

The concepts TIME, PRICE, VOLUME and INQUIRY are to general and thus cannot be associated unambiguously to process map elements. More specific terms (second column) that are potentially available in the categories 'Aktienhandel' and 'Vertrieb' are retrieved in the next step. The term 'Anfrage' did not find a match. The third column displays the conceptualized form of the specific term.

Zeitpunkt	->	`Aktienverkaufszeitpunkt`	->	SHARE_SELLING_TIME
Preis	->	`Aktienverkaufspreis`	->	SHARE_SELLING_PRICE
Volumen	->	`Aktienverkaufsvolumen`	->	SHARE_SELLING_VOLUME

Alternatively, if a technical term model describes that the term 'Aktienverkaufspreis' is an attribute of the term 'Aktiendaten', the retrieval step in the process map element 'Vertrieb' can be avoided because the association is then explicitly described.

As a last step, the terms belonging to the subclass '*ModifierConcept*' are retrieved:

echtzeit	->	REAL_TIME	->	`soaCapability = true`
letzten	->	LAST	->	`soaCapability = false`

Only one of the two modifiers ('echtzeit') can be a non-functional requirement. Potential naming variants due to grammatical suffixes ('echtzeitfähig', 'echtzeitfähigkeit') are detected due to a common stem form.

Further wizard steps allow to manually complete the service description by using the vocabulary available in the KB (see the figure 6.30). Descriptors related to activities are resulting in functional capabilities. In the next wizard step (see 6.31), all relations available in the KB between the descriptors and activities (relation '*associatedTo*') are displayed. In the service description, the noun 'Anfrage', which could not be mapped to a specific descriptor, occurs in the same sentence as the term 'Aktiendaten'. In the KB, the term 'Anfrage' is related to the activity 'anfragen' (relation *hasCorrespondingFunction*) which in turn has a relation '*associatedTo*' to the term 'Aktiendaten'. It is why the wizard step displays this association per default to generate a functional capability called 'Aktiendaten anfragen' with help of a KM lookup in the category '*BObjectBFunction*'.

9.2.2 Interpreting and Linking Process Documentation

The objective of written process documentation is to describe enterprise processes in natural language. Process documentation can be stored in different formats, but the point of interest is the natural language content of the handled documents. To overcome different file format issues and thus avoid the programming of a file reader function for each coexisting one, the SE-BPM engine requires a unique input file format that serves as well as an output format if necessary: XLIFF.

9.2.2.1 XML Localization Interchange File Format

The XML Localization Interchange File Format (XLIFF) is an OASIS standard used to exchange localization data. The current version 1.2 of the XLIFF specification [100] was released in February 2008. XLIFF is comprised in the Open Architecture for XML Authoring and Localization (OAXAL) reference architecture [101], which provides a comprehensive, efficient and cost-effective model regarding the authoring and translation aspects of XML publishing. The objective is to standardize localization processes. OAXAL encompasses key open standards like XML and UNICODE and describes processes which serve as a reference to localization service providers (LSP).

LSPs have to deal with many different file formats and this makes it difficult to pick out the translation units. XLIFF is a standardized format that prepares the input for localization tasks. Rodolfo M. Raya [102] gives an overview of XLIFF and its usage within the translation process. XLIFF requires a map between the original document and XLIFF and the corresponding inverse map. The XLIFF syntax provides following main elements:

- *file*: tag which references the original source document.
- *trans-unit*: tag which stores the text units to translate.
- A set of attributes to store metadata like the hit rate for translation memory systems.

Choosing exactly this format and not another one brings one of the two following advantages:

- If the documentation translation process involves the XLIFF format and the process documentation is intended to be translated, the XLIFF input for translation may already be available.
- If the XLIFF input is not yet available, generating it from the different process documentation files prepares the translation process. The documentation is then ready for translation.

9.2.2.2 Handling Process Documentation

Different use cases built upon one another are involving process documentation:

- 'Document Classifying' (see section 6.2.5.2), which links process map elements or even business processes to documents.
- 'Document Mining' (see section 6.2.5.3), which proposes the generation of business objects according to the text content. Terms which are not part of the KB are proposed for registering.
- 'Document Alignment' (see 6.2.5.4), which compares the process documentation to a process model and detects missing objects at both sides.

To extract the information contained in the process documentation, each translation unit of the XLIFF input file is sent to the first annotator steps of the SE-BPM engine: LE annotator, homogenization annotator and concept annotator. Consumer components interpret the result of the cited annotation chain, depending on each use case.

Following procedure applies for the use case 'Document Classifying':

1. All concepts belonging to a subclass of '*AbstractObjectConcept*' are extracted (only nouns and NPs) to find out which business objects are the point of interest in the process documentation. Terms which were not conceptualized were ignored.

2. The extracted concepts are associated to process map elements in order to classify the concepts in business aspects. This step will only be successful for specific terms.

3. A statistic procedure is performed, counting in which process map element the maximum of terms occurs.

4. If the result is ambiguous (several process map elements with an analog hit count), unspecific terms are implicated. For each previously matched map element, terms deriving from the unspecific terms are searched.

5. The statistic procedure is performed again, counting in which process map element the maximum of terms occur, including the derived terms.

For the use case 'DocumentMining', the annotator chain is completed by a subsequent annotator, the 'reserved words annotator'. This annotator gathers words which are not part of the KB, but give information about the sentence structure (condition, action) and the resulting process structure. Some of the reserved words are analog to the reserved words describing the structure of a software function in programming languages. Additional elements give details about time. This list can be extended on demand. The following list gives an extract of the reserved words with their corresponding German words.

- IF: 'falls', 'wenn', 'in diesem Fall', 'in dem Fall', 'im Falle', 'wenn auch'...
- THEN: 'dann', 'in Folge', 'als Folge'...
- OR: 'oder', 'entweder'
- AND: 'und', 'sowohl ... als auch'
- BEFORE: 'bevor', 'vor', 'vorher', 'früher'...
- AFTER: 'daher', 'hinterher', 'nachher', 'nachdem'...

Following procedure applies for this use case:

1. The 'reserved words annotator' is performed.

2. Words which were not conceptualized are proposed as a term request to the business content expert.

3. Potential IT systems, organizational units and personal names are detected within unknown words depending on the usage context.

4. The sentences or sentence parts are interpreted according to the occurring concepts (all, not only nouns) and reserved words. Business objects are proposed to generation.

5. Sentences which could not be interpreted due to their complexity are marked and signalized to the business modeler.

The 'Document Alignment' use case is based on the results of the 'Document Mining' use case and is a refinement of the 'Document Classifying' use case: on the hand, the results of this procedure can be used to associate a document to a process model if a given percentage of business objects is corresponding to the process document. On the other hand, if a process document is already associated to a process model, the objects proposed for generation in the previous step are compared to the objects occurring in this process model in order to detect misalignments. Following procedure applies:

1. Business objects proposed for generation in the previous step which are not part of the process model are listed.

2. Business objects which occur in the process model and do not correspond to one of the generated objects are listed.

3. Some of the sentences which could not previously be interpreted are compared to the business objects listed in the second step. If the same concepts or concept variants occur, the objects are supposed to be correctly aligned. The second list is then refined.

The procedures of the first two use cases are highlighted by the following example. The text is an extract of a process documentation concerning a training management IT system named 'TrainSys'. This example was already used to illustrate the 'Document Mining' use case in the figure 6.42.

> *Der Schulungskatalog muss angefordert werden.*
> *...*
> *Die ausgedruckte Liste mit dem Schulungsbedarf gibt Auskunft, wer an dieser Schulung teilnehmen soll.*
> *TrainSys updaten.*
> *Die Schulungsbedarfsliste ist jetzt erweitert und ermittelt. Nachdem TrainSys geupdatet wurde, kann der Zugriff neu erfolgen. Der oder die Verantwortliche für die Schulungsorganisation erhält jederzeit alle relevanten Informationen auf Mausklick - entweder als Abfragetabelle oder als formatierten Bericht.*

For the first use case 'Document Classifying', following terms belonging to the subclass '*AbstractObjectConcept*' (first column) are conceptualized (second column) and are associated to a process map element (third column).

Schulungskatalog	->	TRAINING_CATALOG	->	Personalentwicklung
Schulungsbedarf	->	TRAINING_REQUIREMENT	->	Personalentwicklung
Schulungsbedarfsliste	->	TRAINING_REQUIREMENT_LIST	->	Personalentwicklung
Schulungsorganisation	->	TRAINING_ORGANISATION	->	Personalentwicklung
Schulung	->	TRAINING	->	Personalentwicklung

Following terms belonging to the subclass '*AbstractObjectConcept*' (first column) are conceptualized (second column). These terms, all unspecific, were not associated to a process map element.

Liste	->	LIST
Auskunft	->	INFORMATION
Verantwortliche	->	RESPONSIBLE
Zugriff	->	ACCESS
Informationen	->	INFORMATION
Bericht	->	REPORT

All the objects listed in the first list are belonging to the process map element 'Personalentwicklung' (human resource development) so that the result is non-ambiguous. The terms of the second list are not considered in this case and the document is classified under the path 'Support processes -> Human resources -> Human resource development'. To associate this process document to a specific process model or a process step, the same procedure as for the 'Document Alignment' is required.

In the first step of the 'Document Mining' use case, the 'reserved words annotator' marks following words:

und	->	AND
Nachdem	->	AFTER
entweder	->	OR
oder	->	OR

Following terms belonging to the subclass '*AbstractObjectConcept*' do not have an entry in the KB. They will be proposed as a term request to the business content expert.

Abfragetabelle
Mausklick
TrainSys

The sentence '*TrainSys updaten*' and the sentence part '*Nachdem TrainSys geupdatet wurde*' contains the concepts 'TO_UPDATE' linked to an unknown word. As 'TO_UPDATE' can be associated to IT systems, which are mostly not part of natural language elements, 'TrainSys' is supposed supposed to be an IT system.

The next step consists of the generation of business objects. Following business objects are proposed for each sentence:

- *Der Schulungskatalog muss angefordert werden.*: 'Schulungskatalog anfordern' because only two concepts are occurring in this sentence.
- *Die ausgedruckte Liste mit dem Schulungsbedarf gibt Auskunft, wer an dieser Schulung teilnehmen soll.*: this sentence contains too many information units to be correctly interpreted.
- *TrainSys updaten.*: the function object 'TrainSys aktualisieren' is proposed because the preferred form of 'TO_UPDATE' is 'aktualisieren'.
- *Die Schulungsbedarfsliste ist jetzt erweitert und ermittelt.*: this sentence results in two event objects: 'Schulungsbedarfsliste erweitert' and 'Schulungsbedarfsliste ermittelt' because the reserved word 'AND' links to activities related to the same business term. The tense used in this sentence gives an indicator that event objects must be generated.
- *Nachdem TrainSys geupdatet wurde, kann der Zugriff neu erfolgen.*: this sentence is divided in two parts, the conditional part and the action part. The conditional part is recognized by the reserved word 'Nachdem' ('AFTER'), which supposes that an event object must be generated, 'TrainSys aktualisiert'. The action part could not be mapped to a business object because the word 'Zugriff' was too unspecific.
- *Der oder die Verantwortliche für die Schulungsorganisation erhält jederzeit alle relevanten Informationen auf Mausklick - entweder als Abfragetabelle oder als formatierten Bericht.*: this text part could not be interpreted unambiguously due to the sentence complexity. Anyway, the concepts can be used to be compared to existing objects in the 'Document Alignment' use case.

The 'Document Alignment' use case would now compare the generated objects and the ignored sentences to a given process model. This step is not highlighted here because it is intuitive.

Maybe not all the document content should be mapped to business objects. It would make sense to have a semi-structured document whose format is adapted to process documentation. Specific tags could highlight process steps or even condition and action parts. Only the relevant parts should then be transformed into the XLIFF document, which would be a subset of the XLIFF document sent to translation. The output of the 'Document Mining' use case may not be interactive. In case of a batch process, the XLIFF target tag (used to store a translation result) can be used to write the results in a specific format, for instance the list of generated business objects if succeeded or the list of concepts involved in the sentence if the sentence could not be completely processed.

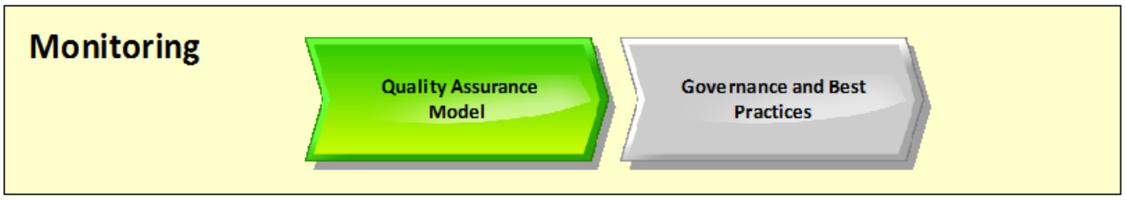
As a conclusion, one can say that the quality of the document mining and alignment use cases hardly correlates with the quality of the process documentation in terms of correctness, appropriateness and readiness. The process documentation should be checked against readiness rules like known from technical documentation systems 'CLAT' or 'Acrocheck'. To reach an optimal result, the terminological resources used for checking the documentation quality must be aligned with the KB content.

This chapter described techniques and supporting software components for exploiting structured and unstructured enterprise resources. These tools are amending the software components needed for checking the quality of business objects stored in business repositories. All these components compose the core of the quality management framework. The next chapter describes metrics and best practices which enable the monitoring of business repositories content.

Chapter 10

Monitoring Business Repositories

The quality assurance framework provides services which check the quality of the objects stored in the business repository. The checks are based on defined rules and metrics which monitor the quality of both business objects and underlying business knowledge models. The deep manual and automatic analysis of the corpus business repository emphasized typical naming errors during the process modeling. The identified errors coupled with existing basic guidelines are summarized in a quality assurance model which is an integral component of the quality assurance framework. The rule set corresponds to errors which really occurred. The model is extensible if required, but each rule necessitates its own implementation. The model compliance is assured by the SE-BPM engine and the applications built on top of the engine. The quality assurance model is amended by recommendations that support a successful roll-out of SE-BPM within the enterprise.



10.1 Quality Assurance Model

The quality assurance model consists of a set of recommendations that describe the correct structure of a business expression for each handled object type and a set of rules that ensure the respect of the naming recommendations. Each rule is described with:

- An external rule number which favors communication or quick references like table entries.
- A title which serves as an official description and can be used as a default error message in SE-BPM related applications.

- A more specific internal error code, used by the SE-BPM engine and applications.
- A status which can be 'Critical' or 'Warning'. Critical errors prevent from further processing. Anyway, the QA-model recommends to correct warning errors as well in order to directly avoid the proliferation of inconsistencies at business object creation time.
- An action which gives some recommendations in order to support the correction process of the business expression.
- A proposition part that explains which kind of proposition is suggested, depending on the rules. It can concern only single words or the whole business expression if the erroneous one can be mapped to a KM entry. Sometimes, no proposition is generated at all due to an ambiguous business expression.
- Examples which compare an incorrect business expression to a correct one. In some examples, several errors may occur. The correct example concentrates on the current error and does not always correct the other ones.

The quality assurance model is divided in 'Correctness Assurance', 'Appropriateness Assurance', 'Readiness Assurance' and 'Coherence Assurance'. The first two categories are common to all the considered object types. The 'Readiness Assurance' is composed of some common rules and completed by business object-specific rules. The 'Coherence Assurance' applies on all object types, but according to their structure not all of them are concerned by all rules. The quality assurance model is supplemented by guidelines that support the governance level to ensure quality criteria of the KBs and KMs.

The quality assurance model introduces some error codes that were not explicitly treated in the chapter 8. While some of the rules are language-independent, others are specific to German. Not all rules are fired for both controlled modeling and restricted modeling, but once the business object was named correctly, they are in both cases compliant with the quality assurance model.

10.1.1 Correctness Assurance

Correctness is the first requirement of the quality assurance model: if a term is not recognized by the LE and is not part of the KB (rule C1 and C2), the business expression cannot be correctly and entirely interpreted. The following rules are less critical because the correct term form can be identified and the subsequent processes can be performed. The correctness assurance concerns each single word that occurs in a business expression.

C1 This term is not part of the common German language.

Error code:	CORR_UNKNOWN
Status:	Critical
Action:	1. If really erroneous, the unknown term must be corrected.

	2. Anglicisms must be replaced by their German equivalents. 3. If the term is correct, but not recognized, it must be requested as a term.
Propositions:	Fuzzy match against the KB term entries so that only terms occurring in the KB are proposed.

Erroneous example	Correct example
Inzahlungsnahme	Inzahlungnahme
Incentive	Leistungszuzahlung

C2 Avoid abbreviations unless they are established acronyms.

Error code:	CORR_ABBR
Status:	Critical
Action:	The abbreviation must be replaced by the full-written term.
Propositions:	No alternative is proposed.

Erroneous example	Correct example
Produktionsauft.	Produktionsauftrag
BR	Betriebsrat
Str. Verk. Ordnung	StVO

C3 This term does not fit the current language and spelling settings.

Error code:	CORR_SPELLING_VARIANT
Status:	Warning
Action:	Spelling variants due to country settings or spelling reforms must be adapted to configured writing conventions.
Propositions:	Corrections recommended by the LE are proposed.

Erroneous example	Correct example
Prozeßprüfung	Prozessprüfung
Abschlußdokumentation	Abschlussdokumentation

C4 Avoid the usage of anglicisms.

Error code:	CORR_ENGL
Status:	Warning
Action:	Anglicisms must be replaced by their German equivalent.
Propositions:	No alternative is proposed, unless the German equivalent is known by the appropriateness check. In this case, the error does not occur.

Erroneous example	Correct example
Process	Prozess
Produkt-Owner	Produkteigentümer

C5 Avoid the usage of hyphens in compound terms, except for separating anglicisms or abbreviations.

Error code:	CORR_HYPHEN
Status:	Warning
Action:	The misplaced, missing or superfluous hyphens must be corrected.
Propositions:	The citation form provided by the LE is proposed.

Erroneous example	Correct example
Fern-Verkehrstour	Fernverkehrstour
KfZSteuer	KfZ-Steuer
Poolfahrzeug	Pool-Fahrzeug

C6 Do not use dashes as a term completion.

Error code:	CORR_COMPL_HYPHEN
Status:	Warning
Action:	The dash must be replaced by the full word. It often leads to term coordinations that are not allowed by readiness rules.
Propositions:	The citation form provided by the LE is proposed.

Erroneous example	Correct example
Fach- und DV-Konzept	Fachkonzept und DV-Konzept
Finanz- & Rechnungswesen	Finanzwesen & Rechnungswesen

10.1.2 Appropriateness Assurance

While correctness assurance ensures the respect of general writing rules, the appropriateness assurance ensures the correct usage of business terminology. The KB is there the authoritative resource, it means that in some cases the content of the KB may overwrite the rules given by the correctness assurance. The general recommendation here is only to respect the terminology stored in the KB.

A1 Avoid unspecific terms.

Error code:	TERM_UNSPECIFIC
Status:	Critical
Action:	The concerned term must be replaced by a more specific one.
Propositions:	Terms deriving from the unspecific ones are proposed. Terms occurring in the same process map element as the modeled process are emphasized.

Erroneous example	Correct example
Daten	Kundendaten
Auftrag	Produktionsauftrag

A2 Use the full-form corresponding to this abbreviation.

Error code:	TERM_ABBR
Status:	Critical
Action:	The abbreviation must be replaced by the full-written term.
Propositions:	The preferred form corresponding to the abbreviation is proposed. The business modeler may have meant a different abbreviation. It is why this rule is considered as critical.

Erroneous example	**Correct example**
BS	Betriebssystem
HR	Personalwesen

A3 Avoid terms including a redundant meaning.

Error code:	TERM_DOUBLE
Status:	Critical
Action:	The concerned term must be replaced by the appropriate term.
Propositions:	Terms related to the inappropriate term are proposed.

Erroneous example	**Correct example**
Auditprüfung	Audit

A4 This term seems to be a synonym of a registered concept. Use the equivalent term instead if it applies.

Error code:	TERM_SYN, derived from the annotation GOV_VARIANT
Status:	Critical
Action:	If the marked term is equivalent in the meaning to the proposed one, it must be replaced. Otherwise, the marked term must be be requested as a new term.
Propositions:	The preferred form corresponding to the guessed synonym term is proposed.

Erroneous example	**Correct example**
Fern-Verkehrstour	Fernverkehrstour
KfZSteuer	KfZ-Steuer
Poolfahrzeug	Pool-Fahrzeug

A5 This term is deprecated. Use the equivalent preferred form instead.

Error code:	TERM_DEPR
Status:	Warning
Action:	The deprecated term must be replaced by the preferred one.

Propositions:	The preferred form corresponding to the deprecated term is proposed.

Erroneous example	**Correct example**
Betriebsversammlungszeitpunkt	Betriebsversammlungstermin
Auftraggeberdaten	Kundendaten

A6 This term is a writing variant from the preferred form. Use the equivalent preferred form instead.

Error code:	TERM_VARIANT
Status:	Warning
Action:	The term variant must be replaced by the authoritative one.
Propositions:	The writing variant as it occurs in the KB is proposed.

Erroneous example	**Correct example**
Betriebsversammlungs-Termin	Betriebsversammlungstermin
erfaßen	erfassen

10.1.3 Readiness Assurance

In contrary to the first two categories, the readiness assurance checks not only single words or noun phrases, but verifies the whole structure of the business expression. Most of the structure errors are critical errors because the business expression can often be rewritten with several possibilities. In some cases, two different objects must be used to formulate the same expression or the model structure may even be altered.

10.1.3.1 Common Readiness Guidelines

R0 The structure of the business expression is too complex. Reformulate it.

Error code:	STRUCT_STRUCTURE
Status:	Critical
Action:	This rule is a fallback rule when the structure could not be recognized. The business modeler must simplify the object denomination.
Propositions:	No business expression is proposed.

Erroneous example	**Correct example**
Aktion Eingangsbelege prüfen auswählen	Aktion 'Eingangsbelege prüfen' auswählen
Fehlende Dokumente früherer Phasen anfordern und anschließend archivieren	?

R1 The usage of the coordination '/' is forbidden. If necessary, rewrite the expression using two separate objects.

Error code:	STRUCT_SLASH
Status:	Critical
Action:	1. Objects containing '/' can be transformed in two separate objects, if necessary connected with an AND or OR operator. 2. In some cases, the two coordinate terms are similar and one of them can be omitted.
Propositions:	No business expression is proposed.

Erroneous example	**Correct example**
Kundenproblem analysieren/klassifizieren	Problem analysieren Problem klassifizieren

R2 The usage of the coordination 'und' is forbidden. If necessary, rewrite the expression using two separate objects.

Error code:	STRUCT_AND
Status:	Critical
Action:	1. Objects containing 'und' can be transformed in two separate objects, if necessary connected with an AND operator. 2. In some cases, the two coordinate terms are similar and one of them can be omitted.
Propositions:	No business expression is proposed.

Erroneous example	**Correct example**
Kundenproblem analysieren und klassifizieren	Problem analysieren Problem klassifizieren

R3 The usage of the coordination 'or' is forbidden. If necessary, rewrite the expression using two separate objects.

Error code:	STRUCT_OR
Status:	Critical
Action:	1. Objects containing 'or' can be transformed in two separate objects, if necessary connected with an OR or XOR operator. 2. In some cases, the two coordinate terms are similar and one of them can be omitted.
Propositions:	No business expression is proposed.

Erroneous example	**Correct example**
Kollege oder Voicebox erreicht	Kollege erreicht Voicebox erreicht

R4 The usage of the coordination '&' is forbidden. If necessary, rewrite the expression using two separate objects.

Error code:	STRUCT_AMP
Status:	Critical
Action:	1. Objects containing '&' can be transformed in two separate objects, if necessary connected with an AND operator. 2. In some cases, the two coordinate terms are similar and one of them can be omitted.
Propositions:	No business expression is proposed.

Erroneous example	**Correct example**
Kunden- & Branchenprofil erstellen	Kundenprofil erstellen Branchenprofil erstellen

R5 The usage of 'gegebenfalls' or 'ggf.' is forbidden. Model the according conditions instead.

Error code:	STRUCT_GGBF
Status:	Critical
Action:	1. 'gegebenfalls' expresses an action after a condition. Model the condition appropriately. 2. If used as a coordination, the business expression may be rewritten with two different objects. 3. If used as a coordination, the two coordinate terms are similar and one of them can be omitted.
Propositions:	No business expression is proposed.

Erroneous example	**Correct example**
Dokument reviewen ggf. abnhemen	Dokument reviewen Dokument abnehmen
ggf. Abschlussangebot fertigstellen	Abschlussangebot fertigstellen

R6 Avoid parenthetical expression

Error code:	STRUCT_PARENTHESE
Status:	Critical
Action:	The business expression must be rewritten without parentheses. It often leads to term coordinations that are not allowed by readiness rules.
Propositions:	No business expression is proposed.

Erroneous example	**Correct example**
Klassen (DV-Sicht) beschreiben	DV-Sichtklassen beschreiben
Beta-Version (Hilfe) erstellen	Beta-Version und Hilfe erstellen
	Beta-Versionshilfe erstellen

10.1.3.2 Guidelines for Data Objects

Data object labels are formed of terms denoting an object, i.e. simple terms, compound terms and NPs, sometimes modified by an adjective. Due to the simple object structure, no specific additional rules are needed for this object type. The common naming pitfalls available in the analyzed business repository are covered by the common rules, particularly the by the appropriateness guidelines.

10.1.3.3 Readiness Guidelines for Process Steps

The structure of process steps should respect following main recommendations (optional parts are denoted by brackets):

> [object modifier] object [activity modifier] activity
> status object [activity modifier] activity
> object 'LITERAL' activity

RA1 Do not nominalize activities.

Error code:	STRUCT_FCT_SUBST
Status:	Critical
Action:	The activity expressed by a substantive must replaced by a verb.
Propositions:	If a business expression involving the occurring concepts is available in the KB, it is proposed.

Erroneous example	**Correct example**
Prüfen der Freigabegrenzen	Freigabegrenzen prüfen
Dokumentieren des Ergebnisses der Installation	Installationsergebnis dokumentieren

RA2 Do not use subordinate clauses.

Error code:	STRUCT_FCT_SUBORDINATE
Status:	Critical
Action:	The activity must remain at the end of the denomination. The status and the object must be expressed with a compound term or an NP.
Propositions:	If a business expression involving the occurring concepts is available in the KB, it is proposed.

Erroneous example	**Correct example**
Prüfen ob Beschaffunsantrag genehmigt	Beschaffungsantragsgenehmigung prüfen
Prüfen, ob Policy unterschrieben	Vorhandensein der Policy-Unterschrift prüfen
	Policy-Unterschrift prüfen
Prüfen, ob Barzahlung gewünscht wird	Barzahlungswunsch prüfen
	Zahlungsart prüfen
Prüfen, ob Schulungsliste vollständig ist	Vollständigkeit der Schulungsliste prüfen

RA3 Do not introduce the activity followed by a colon.

Error code:	STRUCT_FCT_COLON
Status:	Critical
Action:	Place the activity at the end of the business expression and adapt the remaining terms accordingly.
Propositions:	If a business expression involving the occurring concepts is available in the KB, it is proposed.

Erroneous example	**Correct example**
Prüfen: Stornierung Produktionsauftrag	Produktionsauftragstornierung prüfen
Prüfen: Strafe vorhanden	Vorhandensein der Strafe prüfen

RA4 Do not mix statuses and activities.

Error code:	STRUCT_FCT_MIX
Status:	Critical
Action:	Place the activity at the end of the business expression and express the object and its status as an NP .
Propositions:	If a business expression involving the occurring concepts is available in the KB, it is proposed.

Erroneous example	**Correct example**
Packstücke vorhanden prüfen	Vorhandensein von Packstücke prüfen

RA5 Do not use interrogative forms.

Error code:	STRUCT_FCT_INTER
Status:	Critical
Action:	The interrogative form must be replaced by an allowed business expression structure.
Propositions:	If a business expression involving the occurring concepts is available in the KB, it is proposed.

Erroneous example	**Correct example**
Ist Barzahlung gewünscht?	Barzahlungswunsch prüfen
	Zahlungsart prüfen

10.1.3.4 Readiness Guidelines for Events

The structure of events should respect following main recommendations (optional parts are denoted by brackets):

[object modifier] object [activity modifier] status
[object modifier] object comparator NUMBER [unit]
[object modifier] object comparator modifier

RE1 Do not use nominalizations of statuses.

Error code:	STRUCT_EVT_SUBST
Status:	Critical
Action:	The activity expressed by a substantive must replaced by a verb.
Propositions:	If a business expression involving the occurring concepts is available in the KB, it is proposed.

Erroneous example	**Correct example**
Ende Rechtstreit	Rechtstreit beendet

RE2 Express the negation with 'nicht' instead of 'kein'.

Error code:	STRUCT_EVT_KEIN
Status:	Critical
Action:	The negation mark 'kein' must be replaced by 'nicht'. If the amount '0' is meant, the business expression must be reformulated by including the notion of count.
Propositions:	If a business expression involving the occurring concepts is available in the KB, it is proposed. Otherwise the word 'kein' is eliminated and 'nicht' is inserted in a classical event structure.

Erroneous example	**Correct example**
Keine Genehmigung vorhanden	Genehmigung nicht vorhanden
Keine Genehmigung	Genehmigung nicht vorhanden
Kein Teilnehmer storniert	Teilnehmerstornierungsanzahl = 0

RE3 Do not use the verb 'sein' in events.

Error code:	STRUCT_EVT_SEIN
Status:	Warning
Action:	The verb 'sein' must be eliminated. If a comparator is meant, the business expression must be rewritten by using the '=' operator.
Propositions:	The business expression without the verb 'sein' is proposed. If no status is available, 'sein' is replaced by '='.

Erroneous example	**Correct example**
Produktionsauftrag ist vorhanden	Produktionsauftrag vorhanden
Kundenwunsch ist Barzahlung	Kundenwunsch = Barzahlung
	Zahlungsart = Barzahlung

RE4 Do not use verbs in the future or passive form.

Error code:	STRUCT_EVT_WERDEN
Status:	Critical
Action:	The verb 'werden' must be eliminated. Events are always statuses resulting from a function, expressed in the past form. If necessary, the previous function must be adapted as well.
Propositions:	The business expression without the verb 'werden' is proposed.

Erroneous example	**Correct example**
Personalbedarfsliste wird erstellt	Personalbedarfsliste erstellt
Mitarbeiter wird Führungskraft	Mitarbeiter zur Führungskraft ernannt

RE5 Do not use events containing 'zu'. Events must expressed in the past form.

Error code:	STRUCT_EVT_ZU
Status:	Critical
Action:	The particle 'zu' must be eliminated. Events are always statuses resulting from a function, expressed in the past form. If necessary, the previous function must be adapted as well.
Propositions:	The business expression without the verb 'zu' is proposed.

Erroneous example	**Correct example**
Personalbedarfsliste zu erstellen	Personalbedarfsliste erstellt
Angebot ist zu verhandeln	Angebot verhandelt
	Angebotsverhandlung notwendig

RE6 Do not use the verb 'sollen'. Events must be expressed in the past form.

Error code:	STRUCT_EVT_SOLLEN
Status:	Critical
Action:	The verb 'sollen' must be eliminated. Events are always statuses resulting from a function, expressed in the past form. The verb 'sollen' expresses some future and uncertain actions. If necessary, the previous function must be adapted as well.
Propositions:	The business expression without the verb 'sollen' is proposed.

Erroneous example	Correct example
Rechnung soll erstellt werden	Rechnung erstellt
	Rechnungserstellung notwendig

RE7 Do not use the verb 'müssen'. Events must be expressed in the past form.

Error code:	STRUCT_EVT_MUESSEN
Status:	Critical
Action:	The verb 'müssen' must be eliminated. Events are always statuses resulting from a function, expressed in the past form. The verb 'müssen' expresses some future actions. If necessary, the previous function must be adapted as well.
Propositions:	The business expression without the verb 'müssen' is proposed.

Erroneous example	Correct example
Hersteller muss beteiligt werden	Hersteller beteiligt
	Herstellerbeteiligung notwendig

10.1.3.5 Readiness Guidelines for Process Map Elements

Correct structures for process map elements are nouns denoting a process which can be modified by an adjective. Process map items defined by the standard process map must be preferred, but can be completed, particularly in the depth. Unspecific items which do not denote processes must be avoided or used only as leaf elements (for example 'Juni'). This point cannot be checked automatically with high precision. The 'Status' attribute of the rules is always set to 'Warning' because process map elements are not used for subsequent automation activities.

RM1 Activity names must be avoided and nominalizations used instead.

Error code:	STRUCT_PME_SUBST
Status:	Warning
Action:	The activity name must be replaced by a substantive.

Propositions: If available, a substantive corresponding to the object and including the actvity is proposed.

Erroneous example	**Correct example**
Änderungen protokollieren	Änderungsprotokollierung
Release installieren	Release-Installation

RM2 Do not use the term 'Prozess' unless in the top-level map items.

Error code: STRUCT_PME_PROZESS
Status: Warning
Action: The compound part 'Prozess', 'Prozesse' or any equivalent must be eliminated. The top-level elements are predefined.
Propositions: A substantive without the unwished compound part is proposed in simple cases.

Erroneous example	**Correct example**
Auslieferungsprozess	Auslieferung
Auslieferungsprozess Dienstwagen	Dienstwagenauslieferung

10.1.3.6 Coherence Assurance

Even if the business expression respects all the previous recommendations, it may happen that the business expression parts do not respect the usage restrictions defined in the KB. The coherence assurance ensures that only allowed and undefined relations can be associated. Undefined relations are further sent to the governance level. Relations explicitly described as forbidden relations are marked.

H1 'Durchführung' or 'durchführen' is forbidden.

Error code: COH_DURCHF
Status: Critical
Action: The verb 'durchführen' or the substantive 'Durchführung' must be replaced by an appropriate activity name.
Propositions: If possible, the new activity is derived from the object name like highlighted by the the first example.

Erroneous example	**Correct example**
Benutzeranmeldung durchführen	Benutzer anmelden
Betatestdurchführung	Betatest ausführen

H2 The activity cannot be related to this object.

Error code:	COH_RELATION_ACT_OBJ
Status:	Critical
Action:	An alternative activity must be chosen between the allowed ones. Undefined ones are possible, but must be further validated in the restricted modeling use case.
Propositions:	A list of allowed activities is displayed. Context-specific synonyms are highlighted.

Erroneous example	Correct example
Produktionsauftrag informieren	?

H3 The status cannot be related to this object.

Error code:	COH_RELATION_STATUS_OBJ
Status:	Critical
Action:	An alternative status must be choosen between the allowed ones. Undefined ones are possible, but must be further validated in the restricted modeling use case.
Propositions:	A list of allowed statuses is displayed. Context-specific synonyms are highlighted.

Erroneous example	Correct example
Produktionsauftrag konsistent	?

H4 The activity cannot be related to this modifier.

Error code:	COH_RELATION_ACT_MOD
Status:	Critical
Action:	An alternative modifier must be choosen between the allowed ones. Undefined ones are possible, but must be further validated in the restricted modeling use case.
Propositions:	A list of allowed modifiers is displayed. Context-specific synonyms are highlighted.

Erroneous example	Correct example
Produktionsauftrag richtig stornieren	?

H5 The status cannot be related to this modifier.

Error code:	COH_RELATION_STATUS_MOD
Status:	Critical
Action:	An alternative modifier must be choosen between the allowed

	ones. Undefined ones are possible, but must be further validated in the restricted modeling use case.
Propositions:	A list of allowed modifiers is displayed. Context-specific synonyms are highlighted.

Erroneous example	**Correct example**
Produktionsauftrag richtig storniert	?

H6 The object cannot be related to this modifier.

Error code:	COH_RELATION_OBJ_MOD
Status:	Critical
Action:	An alternative modifier must be choosen between the allowed ones. Undefined ones are possible, but must be further validated in the restricted modeling use case.
Propositions:	A list of allowed modifiers is displayed. Context-specific synonyms are highlighted.

Erroneous example	**Correct example**
erfolgslosen Produktionsauftrag stornieren	?

10.2 Quality Assurance of the Business Knowledge Models

The SE-BPM applications assure the respect of the quality assurance model. Anyway, to optimize the checking process, the business knowledge models must fulfill following quality criteria: exhaustiveness, accuracy, restrictiveness, sustainability and reliability. Specialized checks are available at governance level to support the business content experts with the management of business knowledge models. A set of rules is treated by the SE-BPM engine at runtime or by the term mining procedures (rule names starting by GR). The remaining ones are intrinsic to the business knowledge models (rule names starting by GI): the KBs content is checked against itself, domain-specific KBs against the common KB and user-specific KBs against the domain-specific and common KBs.

Each governance rule is described with:

- An external rule number which favors communication or quick references like table entries.
- A title which serves as an official description and can be used as a default error message in SE-BPM related applications.
- A more specific internal error code, used by the SE-BPM engine and applications.

- A list of qualities the rule ensures.
- An explanation which gives more details about the check.
- An action which gives some recommendations in order to support the correction process or the decision process.
- An optional example which shows a typical case coupled with a potential decision.

GR1 Terms stored in the KB as the preferred form must respect the correctness rules.

Error code:	GOV_CORRECTNESS - specified in more detail by the correctness code
Ensures:	reliability
Explanation:	The terms stored in the KBs are checked against the general correctness assurance rules to avoid writing errors during registering. Especially preferred forms are concerned. Besides spelling errors, it avoids that anglicisms or terms which do not respect the hyphenation rules are preferred forms.
Action:	The business content expert can correct the terms if erroneous or choose another term as the preferred form. In exceptional cases only, terms which do not respect the quality criteria can be approved.

GR2 A term supposed to be a variant of an existing one is not registered in the KB.

Error code:	GOV_VARIANT
Ensures:	exhaustiveness, restrictiveness
Explanation:	The marked term was recognized either as a synonym of a registered term or as a writing variant.
Action:	The business content expert must define whether the term must be registered in the KB under the proposed concept or is not a variant. In the latter case, the term can be stored under a different concept or as a negative KM entry to avoid a new similar processing.

Example	**Possible decision**
Reisekosten-Deckblatt	to register as a writing variant of 'Reisekostendeckblatt'
Prozeßschnittstelle	to register as a writing variant of 'Prozessschnittstelle'
Mietpreiskontrolle	to register as a synonym variant of 'Mietpreisprüfung'

GR3 An unregistered term which fulfills the correctness criteria is not registered in the KB.

Error code:	GOV_NEW_TERM_OK
Ensures:	exhaustiveness
Explanation:	The marked term was recognized by the LE and respects the correctness criteria, but is not registered in the KB.
Action:	The business content expert must define whether or not the term must be registered in the KB (as a new concept or an existing one). In the latter case, the term is stored as a negative KM entry to avoid a new similar processing.

Example	Possible decision
Reisekostenbelegzuordnung	could remain unregistered because 'Zuordnung' is
	an action
Lagerhaltungskennzahlen	to register

GR4 An unregistered term which fulfills the critical correctness criteria is not registered in the KB.

Error code:	GOV_NEW_TERM
Ensures:	exhaustiveness
Explanation:	The marked term was recognized by the LE, but does not respect the correctness rules ranked with the status 'warning' and is not registered in the KB.
Action:	The business content expert must define whether or not the term must be registered in the KB (as a new concept or an existing one) and with which writing variant. If not, the term is stored as a negative KM entry to avoid a new similar processing.

Example	Possible decision
Kennzeichen-Verlustdatum	to register as 'Kennzeichenverlustdatum'

GR5 A new term is potentially found.

Error code:	GOV_NEW_UNKNOWN_TERM
Ensures:	exhaustiveness, reliability
Explanation:	The marked term was not recognized by the LE and is not registered in the KB.
Action:	The business content expert must define whether the term must be registered in the KB under an existing concept, a new concept or whether the term is erroneous. In the latter case, the term is stored as a negative KM entry to avoid a new similar processing.

Example	Possible decision
customized	can be registered as a status, for example as a synonym of 'maßgeschneidert'
geperrt	should be 'gesperrt'

GR6 An undefined relation was used.

Error code:	GOV_UNDEFINED_RELATION
Ensures:	accuracy, restrictiveness
Explanation:	When objects, activities, statuses or modifiers are associated and the relation was not explicitly defined by the KB, the used relation is proposed for validation. The relation kind is not specified here.
Action:	The business content expert must define whether the relation is approved or rejected. In the latter case, the relation is stored as a negative KM entry to avoid a new similar processing.

Example	Possible decision
Produktionsauftrag informieren	relation refused
Produktionsauftrag freigeben	relation accepted

GR7 A new business expression respecting the quality assurance model was used.

Error code:	GOV_KM_ENTRY
Ensures:	restrictiveness, sustainability
Explanation:	A business expression which is not yet registered in the KM was used. As it respects the quality assurance model rules, it is proposed to validation to extend the KM.
Action:	Per default, the business expression is stored in the KM with the '*entryStatus*' set to '*generated*'. The business content expert can later verify and validate the business expression. If necessary, it can be manually corrected.

Example	Possible decision
Produktionsauftrag freigeben	corresponds to the previously accepted relation and is correct: can be validated.

GR8 A business expression with an unknown structure was used.

Error code:	GOV_UNKNOWN_STRUCTURE
Ensures:	reliability
Explanation:	A business expression which could not be interpreted was used.
Action:	The business content expert decides whether the new structure

will be implemented as a new rule and whether this structure is a correct one.

Example	Possible decision
Aktion Eingangsbelege prüfen auswählen	will not be implemented: the quotes are missing and this can not be automatically recognized
Fehlende Dokumente früherer Phasen anfordern und anschließend archivieren	will be implemented as a new, but incorrect, rule.

GI1 Following terms stored under different concepts could be synonyms.

Error code:	GOV_INTR_SYN_TERM
Ensures:	accuracy, restrictiveness
Explanation:	A consistency check is performed to detect potential synonyms. The treated elements of the KBs are compound terms and NPs.
Action:	The business content expert must define whether the detected synonyms are really synonyms or represent different concepts.

GI2 Following concept is associated to several synonym elements.

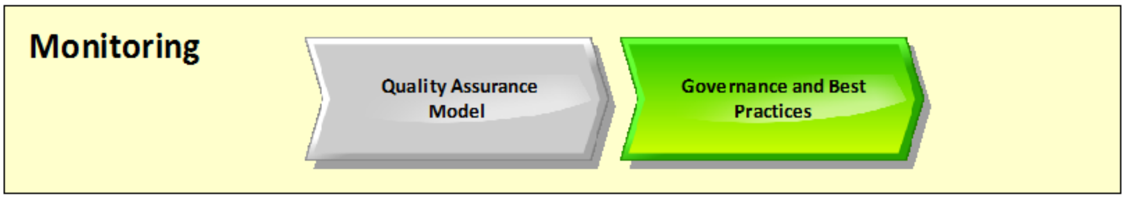
Error code:	GOV_INTR_SYN_RELATION
Ensures:	accuracy, restrictiveness
Explanation:	The relations between objects, activities, statuses and modifiers are checked on concept level. A concept, independently from the concept type, should not be related to context-specific synonyms.
Action:	The business content expert decides whether the concerned concept really needs to be related to the synonym concepts or whether one of them is sufficient.

Example	Possible decision
Produktionsqualität prüfen Produktionsqualität kontrollieren	a relation to the activity 'kontrollieren' is sufficient to express the business expression.

GI3 The usage form of following term issued from the domain-specific or user-specific KB conflicts with the common KB.

Error code:	GOV_INTR_KB_CONFLICT
Ensures:	accuracy, restrictiveness, sustainability reliability
Explanation:	User-specific KBs overwrite domain-specific KBs, which in turn overwrite the common KB. The usage forms of terms are checked to avoid conflicts between the KBs.

Action: The business content expert decides whether the usage form of the more specific KB must overwrite the usage form of the common or less specific KB. In the ideal case, no conflict is available.



10.3 Governance and Best Practices

This section describes the best method companies can apply to roll out SE-BPM applications. Two main initial situations can occur:

- A first introduction of BPM software integrating SE-BPM applications
- Introduction of SE-BPM applications into existing BPM projects

At this point, companies decide whether they manage their process models with controlled modeling or rely on standard knowledge bases only, which costs less administration efforts, but reduces the benefits of SE-BPM applications, particularly when transforming the business process in an IT solution.

10.3.1 Simultaneous Introduction of BPM and SE-BPM Applications

More and more companies discover the benefit of business process management and start new modeling projects supported by BPM software. They can opt for the additional features provided by SE-BPM applications. The advantage in this situation is that no legacy databases are available so that the process modeler can directly start modeling processes with the SE-BPM applications.

10.3.1.1 Introduction of Restricted Modeling

The restricted modeling relies only on the data stored in the standard KB and domain-specific KBs. As a consequence, there is no need for term mining procedures. To support the business modeler during the design tasks, the existing documentation can be classified in the process map. Up to this point, the business modeler can start with the modeling tasks.

10.3.1.2 Introduction of Controlled Modeling

Before starting with the design and the modeling tasks, it is worthwhile to gather enterprise-specific business vocabulary into user-specific KBs by extracting terms from enterprise documentation and terminologies. This optional preparation tasks help to gain time during modeling because the business modeler can directly reuse the business terms without requesting them. After an optional classifying of process documentation into the process map, the process modeler can start with the modeling activities.

10.3.2 Posterior Introduction of SE-BPM Applications

Some companies have modeled their processes for many years and own consequent business repositories. They may be confronted with many of the described issues and want to introduce SE-BPM applications to solve them. The main challenge in this situation is to bring the legacy business repositories in a state which fulfills the requirements of the quality assurance model.

10.3.2.1 Introduction of Restricted Modeling

Preliminary steps are recommended before proceeding with modeling activities in order to fulfill the restricted modeling guidelines:

- Perform a quality check report over the business repositories and correct the marked business objects accordingly.
- Perform a consistency check over the business repositories to detect inconsistencies between business objects which are not covered by the standard and domain-specific KBs. The business objects must be corrected accordingly.
- Optionally classify the process documentation into the process map to support the design process.

Up to this point, the business modeler can continue with his modeling activities by considering the quality assurance model.

10.3.2.2 Introduction of Controlled Modeling

Preliminary steps are recommended before proceeding with modeling activities in order to fulfill the controlled modeling guidelines:

- Perform a quality check report over the business repositories and correct the marked business objects accordingly.
- All business terms occurring in the business repositories which are not part of the standard or domain-specific KBs must be requested and registered in a user-specific KB (business repository mining).
- Perform the governance checks to ensure that the newly registered terms are compliant with the correctness and the governance rules and do not conflict with the underlying KBs.

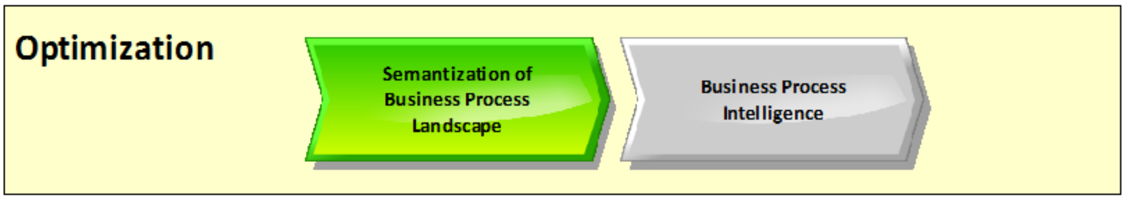
- Perform another quality check report on the business repositories and correct the erroneous business objects involving the previously missing terms.
- Optionally classify the process documentation into the process map to support the design process.
- Gather enterprise-specific business vocabulary into user-specific KBs by extracting terms from enterprise documentation and terminologies.

As the controlled modeling scenario considers all business objects, the consistency check is not needed: all corrected objects are consistent by now. Up to this point, the business modeler can continue with his modeling activities by taking the quality assurance model into consideration.

All other activities like merging processes or transforming business processes into an IT solution do not require preliminary tasks. They directly benefit from a consistent business repository. The business objects that represent the core of all business process modeling related tasks are directly controlled at source level and should at this point respect all the quality assurance model recommendations.

Chapter 11

Conclusion and Perspectives



The dissertation is founded on the hypothesis that the quality of business objects denominations has an impact on the whole Business Process Management life cycle. The analysis performed on the reference and customer business repositories showed that the same business fact can be expressed in several manners due to the usage of synonyms, unspecific terms, forbidden terms or structure of business object denominations that does not fit the modeling guidelines. Additional spelling mistakes or the usage of regional language variants even aggravate the inconsistencies in the business repositories.

The developed quality assurance framework consists of a quality assurance model, software services, methodologies and governance processes, which ensure that the business objects are checked in terms of correctness, appropriateness, readiness and coherence. The use of semantic techniques combined with language technologies allows a semantization of the enterprise process landscape. Adding a semantic layer on top of the business repository enables further operations on a standardized conceptual level. The SE-BPM applications rely on underlying business-specific knowledge models. These models are built by following a bottom-up approach to remain close to the needs and requirements of business modelers. The statistical analysis of business repositories emphasized a set of basic business vocabularies including common synonyms and allowed the development of a method to extract domain-specific business terms. The analysis of business expression structures and the respect of general modeling guidelines resulted in the development of the quality assurance model whose rules are implemented in the SE-BPM engine.

The quality assurance model supported by the SE-BPM engine ensures that the business repositories are growing consistently considering the business object denominations. A consistent business repository allows:

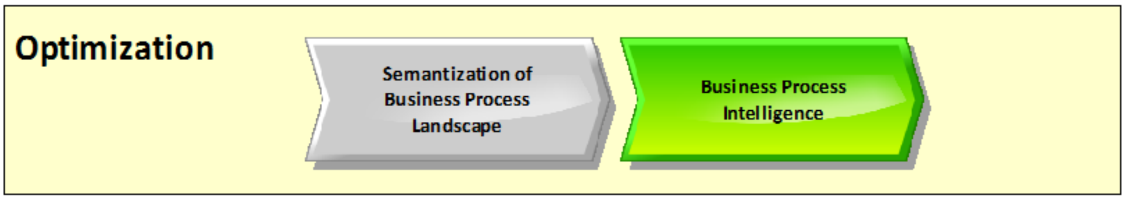
- An efficient retrieval of business objects (findability): finding business objects easily supports the business modeler during modeling time, increases the agility and interoperability of involved process participants and supports change management with an improved impact analysis.

- An effective merging of business processes: an effective merging supports the optimization tasks.

- A seamless transformation into an IT solution: the process execution gains on efficiency with an unified transformation into an IT solution that provides uniform blueprints to software engineers. An efficient services management in a SOA increases the flexibility of the organization because strategic requirements can be fulfilled more intuitively and consistently.

- An off-the-shelf localization of process models: the homogenization of business repository localization tasks reduces the translation costs and increases the interoperability between transcultural process participants.

However, the SE-BPM engine can only bring optimal results if the underlying knowledge models are of high-quality. It means that the entries must be exhaustive, accurate, restrictive, reliable and sustainable. Collaborative work and mining procedures of enterprise resources support the management of knowledge models on governance level.



One objective of the dissertation is to optimize the quality management framework to obtain better process performance. Analog to the domain of business process intelligence, the achieved results must be monitored and evaluated. New terms must continuously be registered and unexpected business expression structures must be considered to reach successive framework optimizations. The machine learning techniques must be improved so that the manual interventions of the business content expert is reduced to the minimum.

Beside the knowledge model and quality assurance rules management, following optimization fields which involve only one language are identified:

- The handled business terms were extracted from the corpus business repositories and did not consider business modeling content standards like ETOM and ITIL. Glossaries issued from such sources are really established in their respective domain. Domain-specific resources should be compliant to them and conflicts with the standard knowledge base must be avoided. Ideally, the standard knowledge base would become a shared and sustainable resource, growing on a collaborative way to establish a common standard set of business terms. This will in fact increase the interoperability not only within one company, but also between companies. Finding the appropriate partner or seamlessly communicate with sub-contractors and customers allows to better react to new market requirements.

- Once a standard knowledge base respecting the qualities of exhaustiveness, restrictiveness, reliability, accuracy and sustainability are implemented, real-life corpus business repositories developed in respect of the quality assurance model can be further analyzed to exploit relations between terms. Performing such statistics on a consistent business repository will bring results of better quality. Several types of relations can be analyzed, for example:

  – Finding out which terms are often related to others in one process model, like the example of shopping websites, when objects are proposed according to the content of the shopping cart. If the business modeler spends less time to collect his needed business terms, the overall modeling time is reduced.

  – Finding out whether activities can be classified in a given order. Taking a computer file as example, the activity of 'creating a file' is the first step prior to 'opening a file'. Reading or writing are logical subsequent actions, while closing and archiving a file are end tasks. It would be interesting to study whether such an intuitive sequence also applies to other or at least to a sufficient amount of business objects. This would reduce the modeling time by generating more precisely business processes, for instance based on a process documentation file.

- The methods for mining enterprise resources focused on data warehouses, process documentation and service descriptions. However, business produces information disseminated in several forms: e-mails, notes, wikis and other collaborative tools. A challenge is to adapt the methods used for interpreting unstructured documentation to detect valuable business information from the various mediums in order to increase the potential of business intelligence tools and in turn improve the capability of taking rapidly the right decisions.

The implemented solution concentrates on the German language, even if multilingual aspects are addressed. Multilingualism has to facets: implementing the solution for other languages or multilingual use cases.

Implementing the solution in a different language implies the following steps:

- Finding a linguistic engine fulfilling the requirements for the concerned language

- Translating the knowledge base and, if necessary, the knowledge matrix into the target language
- Analyzing business repositories in order to:
  - Extracting terms which are not already registered in the translated KB
  - Examining the structure of the business expressions
  - Examining the common pitfalls inherent to the target language
- Defining and implementing according structure rules
- Defining the quality assurance model
- Defining the rules for generating automatically knowledge matrix entries

The easiest language to start with is English if the knowledge base was well defined, the technical names can serve as a default preferred form for each concept. That results in a first version of the knowledge base, but no synonyms are available at this point. Translation tools and utilities which deliver potential synonyms can be involved to support the internationalization process. Coupled with the business repository statistics to obtain a context view for each term, the business content experts are well assisted.

The multilingual use cases are numerous in global companies. To favor the enterprise agility, each business object occurring in the business repository can be translated in different target languages. The technique is analog to the generation of knowledge matrix entries. If the latter are already available, the business object can directly be translated into the target languages at creation time without further intervention of translators. The recommendation is not to translate the business objects, but the knowledge matrix instead. This guarantees a consistent translation and the respect of the quality assurance model. Process models originated from subsidiaries located in different countries and thus potentially modeled in different languages may be merged. If both languages are treated by SE-BPM applications, the merging process can be performed on concept level. An analog use case consists of searching process models modeled in a foreign language with terms entered in the native language. Finally yet importantly, the documentation classification can be extended to documents written in foreign language. If such documents are interesting for process participants of global companies, extracting the main purpose like the involved business objects gives a hint whether they need to be translated or not.

Appendix A

List of implemented Structure Annotator Templates

This appendix contains a list of implemented templates for event objects. Each table row starts with the template name, followed by a few examples of matching business expressions.

`(nicht)_konfiguriert` –> angelegt –> nicht angelegt
`(nicht)_vorhanden` –> möglich –> nicht optional
`(nicht)_iO` –> iO –> nicht ok
`(kein)_(neuen)_antrag` –> neuer Auftrag –> kein Auftrag
`(kein)_(neuen)_antrag_(ist|wird)_(nicht)_angelegt` –> neuer Auftrag ist nicht unterschrieben –> kein Auftrag unterschrieben
`(kein)_(neuen)_antrag_(ist|wird)_(nicht)_vorhanden` –> Keine weiteren Aufträge vorhanden –> Auftrag fehlend
`(kein)_(neuen)_antrag_(ist|wird)_(nicht)_iO` –> neuer Auftrag ist nicht iO –> Auftrag ok
`(kein)_(neuen)_antrag_(ist|wird)_(nicht)_manuell_eingereicht` –> alter Auftrag ist nicht automatisch unterbrochen
`(kein)_(neuen)_antrag_(ist|wird)_(nicht)_ok_eingereicht`

<table>
<tr><td>–> kein Auftrag iO fertiggestellt</td></tr>
<tr><td>(kein)_(neue)_email_(ist|wird)_(nicht)_an_kunden_(ist|wird)_(nicht)_weitergeleitet
–> kein neuer Brief wird an Kunden geschickt
–> Provision ist an Autohändler überwiesen</td></tr>
<tr><td>kunde_hat_(nicht)_bestellt
–> Kunde hat bestellt</td></tr>
<tr><td>kunde_hat_(kein)_empfang_(nicht)_quittiert
–> Kunde hat Empfang quittiert</td></tr>
<tr><td>kunde_moechte_(kein|nicht)_angebot
–> Eingang korrigierte Eingangsrechnung</td></tr>
<tr><td>(kein)_(neuen)_antrag_und|oder|slash_(neue)_rechnung
–> Übergreifende Personalbeschaffung/Personalmarketing
–> Keine Anfragen/ Aufträge</td></tr>
<tr><td>(kein)_(neuen)_antrag_und|oder|slash_(neue)_rechnung_(ist|wird)_(nicht)_gewuenscht
–> Material/Dienstleitung ist geliefert</td></tr>
<tr><td>(kein)_(neuen)_antrag_und|oder|slash_(neue)_rechnung_(ist|wird)_(nicht)_fehlerhaft
–> Finanzierungsvertrag/-unterlagen sind vorhanden</td></tr>
<tr><td>(kein)_(neuen)_antrag_und|oder|slash_(neue)_rechnung_(ist|wird)_(nicht)_iO
–> Finanzierungsvertrag/-unterlagen sind i.O.</td></tr>
<tr><td>(kein)_(neue)_unterlagen_(sind)_(nicht)_blau_(und|oder|slash)_(gelb)
–> Ampel rot oder gelb</td></tr>
<tr><td>(kein)_(neue)_unterlagen_(sind)_(nicht)_richtig_(und|oder|slash)_(vollstaendig)
–>Anfrage nicht erfüllbar oder unwirtschaftlich</td></tr>
<tr><td>(kein)_(neue)_antrag_(ist|wird)_(nicht)_zu_(kreiren)
–> Angebot ist zu modifizieren
–> Kein neues Angebot zu erstellen</td></tr>
<tr><td>(schaden)_(nicht)_zu_(pruefen)_[falsedisam]
–> Schaden zu prüfen
Note: Matches if the first noun is analyzed as a verb by the LE</td></tr>
<tr><td>(kein)_(blaue)_rechnung_soll_(nicht)_erstellt_werden
–> alte Dokumente sollen archiviert werden</td></tr>
<tr><td>(kein)_(rotes)_auto_(ist)_(nicht)_auf_lager
–> Ähnliches Fahrzeug auf Lager</td></tr>
<tr><td>(kein)_(rotes)_auto_(ist)_(nicht)_in_ordnung
–> eingereichte Daten sind in Ordnung</td></tr>
<tr><td>(unter)_5</td></tr>
</table>

<table>
<tr><td>–> Über 10</td></tr>
<tr><td>(mathematiczeichen)_5_einheit
–> < 2 km</td></tr>
<tr><td>start_evaluation
–> Ende Rechtstreit</td></tr>
<tr><td>eingang_gutschrift_und|oder|slash_rechnung
–> Eingang Brief oder E-Mail</td></tr>
<tr><td>(kein)_np
–> Kein Schein des Finanzamtes</td></tr>
<tr><td>(kein)_(neue)_np_(ist|wird)_(nicht)_angelegt
–> Verbesserungen des Managementsystems umgesetzt</td></tr>
<tr><td>(kein)_(neue)_np_(ist|wird)_(nicht)_vorhanden
–> umfangreiche Mangelmeldung des Auftraggebers vorhanden</td></tr>
<tr><td>(kein)_(neue)_np_(ist|wird)_(nicht)_an_kunden_(ist|wird)_(nicht)_weite rgeleitet
–> Kopie der Unterlagen sind an Kunden ausgehändigt</td></tr>
<tr><td>(kein)_(neue)_email_(ist|wird)_(nicht)_an_np_(ist|wird)_(nicht)_weite rgeleitet
–> Änderungsvorschlag an Antragsteller der Methodenerweiterung gesendet</td></tr>
<tr><td>(kein)_(neue)_np_(ist|wird)_(nicht)_an_np_(ist|wird)_(nicht)_weitergel eitet
–> gefunde Grundursache des Problems an Antragsteller der Methodenerweiterung mitgeteilt</td></tr>
<tr><td>(kein)_(neue)_np_liegt_(nicht)_vor
–> neuer Vertragsvorschlag des Kunden liegt vor</td></tr>
<tr><td>kunde_moechte_(kein)_np
–> Auftraggeber wünscht Überarbeitung des Lösungskonzeptes</td></tr>
<tr><td>literal_(ist|wird)_(nicht)_angezeigt
–> ’Suche beendet’ angezeigt</td></tr>
<tr><td>meldung_literal_(ist|wird)_(nicht)_angezeigt
–> Fehlermeldung ’Suche abgebrochen’ angezeigt</td></tr>
</table>

Table A.1: Template lists for event objects

Appendix B

Common LE Output Features

This appendix contains a list of the most common features provided by the german LE that are or can be useful for the implementation of the SE-BPM engine. The LE produces output in form of feature bundles containing a set of attribute/value pairs.

Non-linguistical Features		
Attribute	Values	Description
ori	string	Original word or word group like provided by the input but special characters (like blanks are normalized.
mori	string	Identical to oris, but the normalized characters are escaped. This allows to reconstruct the original word.
saw	string	Separator characters that are occurring after the current word. Useful to reconstruct the original sentence.
snr	integer	Sentence number
wnra	integer	Absolute word number relative to the whole analyzed text.
wnrr	integer	Word number relative to the sentence.
gra	small | cap | caps | digits | other	Describes the graphics of the word (capital letters, digits...)

Linguistical Features		
Attribute	Values	Description
c	noun | verb | adj | adv | w | vpref | quant | fromto | brand | z | week | sgml	Grammatical category.

g	m | f | n	Gender of a noun: masculine, feminine, neuter.
case	nom | gen | acc | dat	Case of a noun: nominative, genitive, accusative, dative.
nb	sg | plu	Number of a noun or a verb: singular or plural.
deg	base | comp | sup	Degree of an adjective.
vtyp	fiv | inf | ptc1 | ptc2 | izu | imperativ	Verb type (ex. infinitive, past participle...).
tns	pres | past	Tense of a verb: present, past.
per	1 | 2 | 3	Denotes if the verb is conjugated in the 1st, 2nd or 3rd person.

Structural Information Features		
Attribute	Values	Description
ts	string	Segmented string (compound parts).
t	string	Segmented string, each segment consists of the lexical unit.
ds	string	Segmented string that contains additional derivation information.
ls	string	Lexical structure that contains additional derivation information.
lu	string	Lexical unit, base form of the word.
zf	string	Citation form, like a dictionary entry.
w	integer	Number of compound parts. If the value is $>$ 10, one or more compound parts are unknown.

Semantic Features		
Attribute	Values	Description
s	string	Semantic classification (ex: agent, money, process, plant, family, colour, music...).
ss	string	Concatenation of semantic values (s). Each value corresponds to a compound part.

List of Figures

List of Tables

Bibliography

[1] Mendling, Jan and Hajo Reijers: *The Impact of Activity Labeling Styles on Process Model Quality.* In *Proceedings 3rd EuroSIGSAND Symposium 2008*, pages 117–127, Marburg/Lahn, June 2008. Koellen-Verlag.

[2] Peters, Nicolas and Matthias Weidlich: *Using Glossaries to Enhance the Label Quality in Business Process Models.* In *EPK 2009 – Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*, pages 75–90, Berlin, November 2009.

[3] Leopold, Henrik, Sergey Smirnov and Jan Mendling: *Refactoring of Activity Labels in Business Process Models.* In *Proceedings 15th International Conference on Applications of Natural Language to Information Systems*, Cardiff, June 2010. Springer.

[4] Leopold, Henrik, Sergey Smirnov and Jan Mendling: *On Labeling Quality in Business Process Models.* In *EPK 2009 – Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*, pages 42–57, Berlin, November 2009.

[5] Mendling, Jan, Hajo Reijers and Jan Recker: *Activity Labeling in Process Modeling: Empirical Insights and Recommendations.* Information Systems, 2009.

[6] Friedrich, Fabian: *Measuring Semantic Label Quality Using WordNet.* In *EPK 2009 – Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*, pages 7–21, Berlin, November 2009.

[7] Peters, Nicolas and Matthias Weidlich: *Automatic Generation of Glossaries for Process Modelling Support.* Enterprise Modelling and Information Systems Architectures, 6(1), February 2011.

[8] Delfmann, Patrick, Sebastian Herwig and Łukasz Lis: *Konfliktäre Bezeichnungen in Ereignisgesteuerten Prozessketten – Linguistische Analyse und Vorschlag eines Lösungsansatzes.* In *EPK 2009 – Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten*, pages 178–194, Berlin, November 2009.

[9] Johansson, Henry J., Patrick McHugh, A. John Pendlebury and William A. Wheeler: *Business Process Reengineering: BreakPoint Strategies for Market Dominance.* John Wiley & Sons, May 1993.

[10] Ivanov, Konstantin: *10 Definitions of Enterprise Architecture. Which corresponds to Yours?* URL: http://www.ariscommunity.com/users/koiv/2009-08-20-10-definitions-enterprise-architecture-which-corresponds-yours, August 2009.

[11] The Open Group: *Welcome to TOGAF Version 9 'Enterprise Edition'.* URL: http://www.opengroup.org/togaf/, 2010.

[12] The Department of Defense: *The DoDAF Architecture Framework Version 2.02*. URL: http://cio-nii.defense.gov/sites/dodaf20/index.html.

[13] The Ministry of Defence: *MOD Architecture Framework (MODAF)*. URL: http://www.mod.uk/DefenceInternet/AboutDefence/WhatWeDo/InformationManagement/MODAF/.

[14] Jeanne W. Ross, Peter Weill, David C. Robertson: *Enterprise Architecture As Strategy: Creating a Foundation for Business Execution*. Harvard Business Press, June 2006.

[15] Members of ADiWa project: *ADiWa – Vom Internet der Dinge zu intelligenten Geschäftsprozessen*. URL: http://www.adiwa.net/fileadmin/templateadiwa/ADiWa_Infoblatt.pdf, Januar 2009.

[16] OASIS: *Reference Model for Service Oriented Architecture 1.0*. URL: http://docs.oasis-open.org/soa-rm/v1.0/, October 2006.

[17] Object Management Group: *Documents Associated With CORBA, 3.1*. URL: http://www.omg.org/spec/CORBA/3.1/, January 2008.

[18] Microsoft Corporation: *[MS-DCOM]: Distributed Component Object Model (DCOM) Remote Protocol Specification*. URL: http://msdn.microsoft.com/library/cc201989.aspx, 2007-2008.

[19] Arsanjani, Ali, Grady Booch, Toufic Boubez, Paul C. Brown, David Chappell, John deVadoss, Thomas Erl, Nicolai Josuttis, Dirk Krafzig, Mark Little, Brian Loesgen, Anne Thomas Manes, Joe McKendrick, Steve Ross-Talbot, Stefan Tilkov, Clemens Utschig-Utschig and Herbjörn Wilhelmseni: *SOA Manifesto*. URL: http://www.soa-manifesto.org/, 2009.

[20] W3C: *Web Services Description Language (WSDL) Version 2.0 Part 0: Primer*. URL: http://www.w3.org/TR/wsdl20-primer/, June 2007.

[21] W3C: *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*. URL: http://www.w3.org/TR/wsdl20/, June 2007.

[22] W3C: *Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts*. URL: http://www.w3.org/TR/wsdl20-adjuncts/, June 2007.

[23] Berners-Lee, Tim, Roy T. Fielding and Larry Masinter: *Uniform Resource Identifier (URI): Generic Syntax*. URL: http://labs.apache.org/webarch/uri/rfc/rfc3986.html, January 2005.

[24] Berners-Lee, Tim: *HyperText Transfer Protocol*. URL: http://www.w3.org/History/19921103-hypertext/hypertext/WWW/Protocols/HTTP.html, 1992.

[25] W3C: *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*. URL: http://www.w3.org/TR/soap12-part1/, April 2007.

[26] Bruijn, Jos de, Christoph Bussler, John Domingue, Dieter Fensel, Martin Hepp, Uwe Keller, Michael Kifer, Birgitta König-Ries, Jacek Kopecky, Rubén Lara, , Holger Lausen, Eyal Oren, Axel Polleres, Dumitru Roman, James Scicluna and Michael Stollberg: *Web Service Modeling Ontology (WSMO)*. URL: http://www.w3.org/Submission/WSMO/, June 2005.

[27] ESSI WSMO Working group: *WSMO Ontologies*. URL: http://www.wsmo.org/.

[28] Bruijn, Jos de, Dieter Fensel, Uwe Keller, Michael Kifer, Holger Lausen, Reto Krummenacher, Axel Polleres and Livia Predoiu: *Web Service Modeling Language (WSML)*. URL: http://www.w3.org/Submission/WSML/, June 2005.

[29] Object Management Group: *MDA: The Architecture of Choice for a Changing World*. URL: http://www.omg.org/mda/executive_overview.htm, August 2010.

[30] Object Management Group: *MDA Guide Version 1.0.1*. URL: http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf, June 2003.

[31] Object Management Group: *OMG Unified Modeling Language (OMG UML), Infrastructure version 2.3*. URL: http://www.omg.org/spec/UML/2.3/Infrastructure/PDF/, May 2010.

[32] Object Management Group: *OMG Unified Modeling Language (OMG UML), Superstructure version 2.3*. URL: http://www.omg.org/spec/UML/2.3/Superstructure/PDF/, May 2010.

[33] Object Management Group: *Meta Object Facility (MOF) Core Specification*. URL: http://www.omg.org/spec/MOF/2.0/PDF, January 2006.

[34] ISO: *ISO/IEC 19503:2005 Information technology - XML Metadata Interchange (XMI)*. URL: http://www.iso.org/iso/catalogue_detail.htm?csnumber=32622, October 2010.

[35] Guttman, Michael and John Parodi: *Real-life MDA: solving business problems with model driven architecture*. The MK/OMG Press, May 2007.

[36] O'Reilly, Tim: *What Is Web 2.0 – Design Patterns and Business Models for the Next Generation of Software*. URL: http://oreilly.com/web2/archive/what-is-web-20.html, September 2005.

[37] O'Reilly, Tim and John Battelle: *Web Squared: Web 2.0 Five Years On*. URL: http://assets.en.oreilly.com/1/event/28/web2009_websquared-whitepaper.pdf, October 2009.

[38] McAfee, Andrew P.: *Enterprise 2.0: The Dawn of Emergent Collaboration*. URL: http://sloanreview.mit.edu/the-magazine/articles/2006/spring/47306/enterprise-the-dawn-of-emergent-collaboration/, April 2006.

[39] McAfee, Andrew P.: *Enterprise 2.0: New Collaborative Tools for Your Organization's Toughest Challenges*. Harvard Business Publishing, December 2009.

[40] Howe, Jeff: *The Rise of Crowdsourcing.* URL: http://www.wired.com/wired/archive/14.06/crowds.html?pg=1&topic=crowds&topic_set=, June 2006.

[41] Twitter: *Twitter Home Page.* URL: http://twitter.com/.

[42] Rosemann, Michael: *Where BPM and Twitter could meet.* URL: http://www.ariscommunity.com/users/mrosemann/2009-08-07-where-bpm-and-twitter-could-meet, August 2009.

[43] *ARIS Community.* URL: http://www.ariscommunity.com/.

[44] *ARISalign.* URL: http://www.alignspace.com/.

[45] *PROWIT.* URL: http://www.prowit-projekt.de.

[46] Walter, Philipp: *From Web 2.0 to Human Centric BPM with PROWIT.* URL: http://m.ariscommunity.com/users/philippwalter/2010-09-28-web-20-human-centric-bpm-prowit, September 2010.

[47] Jørgensen, Håvard: *12 Different Ways to Model Business Processes.* URL: http://activeknowledgemodeling.com/2009/03/31/12-different-ways-to-model-business-processes/, March 2009.

[48] Object Management Group: *Business Process Model and Notation (BPMN).* URL: http://www.omg.org/spec/BPMN/1.2/, January 2009.

[49] White, Stephen A. - IBM Corp.: *Introduction to BPMN.* URL: http://www.bpmn.org/Documents/Introduction_to_BPMN.pdf/, May 2004.

[50] Scheer, August-Wilhelm: *ARIS vom Geschäftsprozess zum Anwendungssystem.* Springer, April 2001.

[51] Scheer, August-Wilhelm: *ARIS - Modellierungsmethoden, Metamodelle, Anwendungen.* Springer, November 1999.

[52] Stein, Sebastian: *EPC vs. BPMN - the perfect flamewar.* URL: http://www.ariscommunity.com/users/sstein/2010-04-15-epc-vs-bpmn-perfect-flamewar/, April 2010.

[53] WmFC: *XPDL Support and Resources.* URL: http://www.wfmc.org/xpdl.html/, 2010.

[54] Shapiro, Robert M.: *XPDL 2.2: Incorporating BPMN 2.0 Process Modeling Extensions.* URL: http://bpmnhandbook.com/04_papers/Shapiro-%20XPDL_2.2_Incorporating_BPMN_2.0_Process_Modeling_Extensions_v2.pdf, 2010.

[55] OASIS: *Web Services Business Process Execution Language Version 2.0.* URL: http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf, April 2007.

[56] Active Endpoints, Adobe Systems, BEA Systems, IBM, Oracle and SAP: *WS-BPEL Extension for People.* http://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/30c6f5b5-ef02-2a10-c8b5-cc1147f4d58c, June 2007.

[57] Active Endpoints, Adobe Systems, BEA Systems, IBM, Oracle and SAP: *Web Services Human Task.* URL: http://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/a0c9ce4c-ee02-2a10-4b96-cb205464aa02, June 2007.

[58] Vignéras, Pierre: *Why BPEL is not the holy grail for BPM.* URL: http://www.infoq.com/articles/bpelbpm, October 2008.

[59] White, Stephen A. - IBM Corp.: *Using BPMN to Model a BPEL Process.* URL: http://www.omg.org/bpmn/Documents/Mapping_BPMN_to_BPEL_Example.pdf, February 2005.

[60] Russell, Nick, Wil M.P. van der Aalst, Arthur H.M. ter Hofstede and Petia Wohed: *On the Suitability of UML 2.0 Activity Diagrams for Business Process Modelling.* In *Proceedings of the Third Asia-Pacific Conference on Conceptual Modelling (APCCM2006)*, Hobart, Australia, 2006.

[61] Porter, Michael E.: *Competitive Advantage: Creating and Sustaining Superior Performance.* Free Press, 1998 (1985).

[62] Chen, Peter Pin-Shan: *The entity-relationship model – toward a unified view of data.* ACM Transactions on Database Systems (TODS), 1, March 1976.

[63] Rose, Kenneth H.: *Project Quality Management - Why, What and How.* J. Ross Publishing, June 2005.

[64] Deming, W. Edwards: *Out of the Crisis.* MIT Press, August 2000.

[65] ISO: *Selection and use of the ISO 9000 family of standards.* URL: http://www.iso.org/iso/iso_9000_selection_and_use-2009.pdf, February 2009.

[66] ISO: *ISO 9000 essentials.* URL: http://www.iso.org/iso/iso_catalogue/ management_and_leadership_standards/quality_management/iso_9000_essentials, 2010.

[67] Wikipedia: *ISO 9000.* URL: http://en.wikipedia.org/wiki/ISO_9000, November 2010.

[68] G-14 Americas Aerospace Quality Standards Committee (Aaqsc): *Quality Management Systems - Requirements for Aviation, Space and Defense Organizations.* URL: http://standards.sae.org/as9100c/, January 2009.

[69] QuEST Forum: *About TL 9000.* URL: http://tl9000.org/, 2010.

[70] ISO: *ISO/TS 16949:2009.* URL: http://www.iso.org/iso/catalogue_detail?csnumber=52844, 2010.

[71] ISO: *ISO 14000 essentials.* URL: http://www.iso.org/iso/iso_catalogue/management_and_leadership_standards/environmental_management/iso_14000_essentials, 2010.

[72] Pyzdek, Thomas and Paul A. Keller: *The Six Sigma Handbook, Third Edition.* McGraw-Hill, September 2009.

[73] Kneller, Maggie: *Executive Briefing: The Benefits of ITIL.* URL: http://www.best-management-practice.com/gempdf/OGC_Executive_Briefing_Benefits_of_ITIL.pdf, September 2010.

[74] Iqbal, Majid and Michael Nieves: *ITIL Service Strategy.* The Stationary Office, May 2007.

[75] Lloyd, Vernon and Colin Rudd: *ITIL Service Design.* The Stationary Office, May 2007.

[76] Lacy, Shirley and Ivor Macfarlane: *ITIL Service Transition.* The Stationary Office, May 2007.

[77] Cannon, David and David Wheeldon: *ITIL Service Operation.* The Stationary Office, May 2007.

[78] Spalding, George and Gary Case: *ITIL Continual Service Improvement.* The Stationary Office, May 2007.

[79] ITIL IT Service Management: *Glossary of Terms, Definitions and Acronyms.* URL: http://www.best-management-practice.com/gempdf/ITIL_Glossary_V3_1_24.pdf, May 2007.

[80] tmforum: *Business Process Framework (eTOM) In Depth.* URL: http://www.tmforum.org/BestPracticesStandards/BusinessProcessFramework/6637/Home.html, 1988-2011.

[81] Misra, Kundan: *OSS for telecom networks: an introduction to network management.* Springer, August 2004.

[82] Supply-Chain Council: *Supply-Chain Operations Reference-model.* URL: http://supply-chain.org/f/SCOR%2090%20Overview%20Booklet.pdf, 2008.

[83] Berners-Lee, Tim: *Linked Data.* URL: http://www.w3.org/DesignIssues/LinkedData, July 2006.

[84] Berners-Lee, Tim: *Giant Global Graph.* URL: http://dig.csail.mit.edu/breadcrumbs/node/215, November 2007.

[85] Berners-Lee, Tim: *Semantic Web - XML2000.* URL: http://www.w3.org/2000/Talks/1206-xml2k-tbl/Overview.html, 2000.

[86] W3C: *Resource Description Framework (RDF): Concepts and Abstract Syntax.* URL: http://www.w3.org/TR/rdf-concepts/, February 2004.

[87] Dublin Core Metadata Initiative: *Dublin Core Metadata Element Set, Version 1.1.* URL: http://dublincore.org/documents/dces/, October 2010.

[88] W3C: *SPARQL Query Language for RDF.* URL: http://www.w3.org/TR/rdf-sparql-query/, January 2008.

[89] W3C: *RDF Vocabulary Description Language 1.0: RDF Schema.* URL: http://www.w3.org/TR/rdf-schema/, February 2004.

[90] Berners-Lee, Tim and Dan Connolly: *Notation3 (N3): A readable RDF syntax.* URL: http://www.w3.org/TeamSubmission/n3/, January 2008.

[91] Beckett, David and Tim Berners-Lee: *Turtle - Terse RDF Triple Language.* URL: http://www.w3.org/TeamSubmission/turtle/, January 2008.

[92] W3C: *RDF Test Cases*. URL: http://www.w3.org/TR/rdf-testcases/, February 2004.

[93] W3C: *OWL 2 Web Ontology Language – Structural Specification and Functional-Style Syntax*. URL: http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/, October 2009.

[94] W3C: *OWL Web Ontology Language – Semantics and Abstract Syntax*. URL: http://www.w3.org/TR/owl-semantics/, February 2004.

[95] Patel-Schneider, Peter F., Bell Labs Research and Lucent Technologies: *A Proposal for a SWRL Extension towards First-Order Logic*. URL: http://www.w3.org/Submission/SWRL-FOL/, April 2005.

[96] W3C: *RIF Core Dialect*. URL: http://www.w3.org/TR/rif-core/, June 2010.

[97] DUDEN: *Crashkurs: In 25 Schritten zur neuen Rechtschreibung*. URL: http://www.duden.de/deutsche_sprache/sprachwissen/rechtschreibung/crashkurs/lautbuchstaben/index.php, 2010.

[98] Klückmann, Jörg: *10 Steps to Business-Driven SOA*. URL: http://www.softwareag.com/de/images/sec_IDS_10Steps_BD-SOA_WP_Oct10_tcm17-80002.pdf, 2010.

[99] Alvestrand, H.: *Tags for the Identification of Languages*. URL: http://www.ietf.org/rfc/rfc1766.txt, 1995.

[100] OASIS: *XLIFF Version 1.2*. URL: http://docs.oasis-open.org/xliff/xliff-core/xliff-core.pdf, February 2008.

[101] OASIS: *Reference Model for Open Architecture for XML Authoring and Localization 1.0*. URL: http://docs.oasis-open.org/xliff/xliff-core/xliff-core.pdf, October 2009.

[102] Raya, Rodolfo M.: *XML Localization Interchange File Format as an intermediate file format*. URL: http://wwww.maxprograms.com/articles/xliff.html, October 2004.

[103] Schütz, Jörg and Alexandra Weissgerber: *Semantically Enriched Business Process Modeling*. In *Proceedings of BITKOM KnowTech 2008*, Frankfurt, Germany, October 2008.

[104] Schütz, Jörg and Alexandra Weissgerber: *Localization, a networked Commodity?* In *LRC XV – Brave New World – The Future of Localisation Services*, Limerick, Irland, September 2010.

[105] Weissgerber, Alexandra and Jörg Schütz: *Localizing Business Process Repository Content*. In *1st XLIFF International Symposium*, Limerick, Irland, September 2010.