Configuration and Change Maturity Optimization

The Case of Complex Engineering Environments

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der Fakultät für Maschinenbau des

Karlsruher Instituts für Technologie (KIT)

Genehmigte

DISSERTATION

M. Sc. Masoud Niknam

Referent: Prof. Dr. Dr.-Ing. Dr. h. c. Jivka Ovtcharova

Korreferent: Prof. Dr. Pierre Bonnal

Tag der mündlichen Prüfung: 15.12.2017

Institute for Information Management in Engineering

Karlsruhe Institute of Technology

Prof. Dr. Dr. Ing. Dr. h.c. Jivka Ovtcharova

Configuration and Change Maturity Optimization

The Case of Complex Engineering Environments

A Doctoral Thesis to Obtain the Degree of

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

For Submission to the Faculty of Mechanical Engineering

Karlsruhe Institute of Technology (KIT)

DISSERTATION

by

Masoud Niknam

Main Examiner: Prof. Dr. Dr. Ing. Dr. h.c. Jivka Ovtcharova

Co-Examiner: Prof. Dr. Pierre Bonnal

Acknowledgements

This dissertation could not have been finished without the several supporters and contributors who have genuinely put their trust and effort in me and my work. I would like to take this opportunity to show my appreciation and gratitude to them.

First and foremost, my “Doktormutter”, Prof. Dr. Dr. Ing. Dr. h.c. Jivka Ovtcharova, who over the course of this work, compassionately provided me with excellent academic support, vast amount of knowledge and experience as well as several professional collaboration networks. Without her consistent encouragements, this work would have not taken off the ground in the first place.

Secondly, I would like to thank Prof. Dr. Bonnal for his ongoing inspirational contributions. His positive energy, rich knowledge, broad vision and generous care have always been my sources of motivation and enthusiasm through every phase of this thesis.

I would also like to appreciate the collaborations that I have received from several academic and professional communities such as my peers at IMI, my colleagues in PURESAFE project and the several bachelor and master thesis students that I got a chance to work with during the course of my research. Their sincere contributions directly and indirectly had significant positive impact on the structure and content of this work.

Last but not least, I would like to express my highest gratitudes to my family. My wife, Soode, who had my back through every single second of this work and made all possible sacrifices for me to accomplish my dream and succeed. She made me believe in myself and my potentials and made me a stronger person day by day; To the most beautiful daughter in the world, Ariana, who eversince decided to step foot in this world, gave me an extraordinary power and motivation to finalize this work; To my parents and my two older brothers, who paved the way for my growth and success at every stage of my life and encouraged me to dream big and work hard to achieve it.

Kurzfassung

Im derzeitigen globalen Wettbewerbsmarkt, in dem Unternehmen durch die Bereitstellung anspruchsvoller, aber kundenorientierter Lösungen einen höheren Marktanteil erreichen, ist das Management und der Schutz von Produktinformationen über verschiedene Phasen seines Lebenszyklus (von der Erfassung der Kundenanforderungen bis hin zum Prototyping und Start) von bedeutende Bedeutung. Die Disziplin, die Sicherheit für das Produktinformationsmanagement bietet, ist das Konfigurationsmanagement und erfordert je nach Umfang und Komplexität der Anwendungsumgebung mehrere Aktivitäten und Maßnahmen, die ordnungsgemäß implementiert werden müssen.

Diese Arbeit zielt darauf ab, einen Optimierungsrahmen für die Industrie zur Verfügung zu stellen, um ihre Kompetenz in dieser Disziplin zu bewerten und ihre Prozesse zu stärken und somit einen Wettbewerbsvorteil unter ihren Konkurrenten zu erlangen. Dieses Ziel wurde erreicht, indem zuerst die kritischen Erfolgsfaktoren des Configuration Managements identifiziert und ein Reifegrad-Modell entwickelt wurde, das die Basisreife von Organisationen im Hinblick auf ihre Konfigurationsmanagementbemühungen bewertet und einen Fahrplan für weitere Verbesserungen liefert. Das entwickelte Reifegradmodell wurde dann durch eine branchenübergreifende Bewertung validiert und die Ergebnisse legten die Grundlage für die Identifizierung des aktuellen Reifegrads in verschiedenen Branchen, Möglichkeiten zur Verbesserung der Verwendbarkeit des Reifegradmodells für Evaluierungszwecke sowie die möglichen Forschungsmöglichkeiten für Lösungen die Verbesserungen in den Prozessfähigkeiten der Disziplin bieten.

Das Hauptpotenzial für die weitere Entwicklung des Modells wurde Durch das Fehlen quantitativer Schlüsselleistungsindikatoren (KPIs) identifiziert, und es wurde durch Identifizierung relevanter KPIs, die eine quantitative Messung der Reife von Organisationen ermöglichen, adressiert. Gleichzeitig wurde eine Möglichkeit zur Konzeption eines Entscheidungshilfesystems für den Prozess des Änderungsprozesses identifiziert, das historische Kenntnisse früherer Konfigurationsänderungen nutzen kann, um die Entscheidungsträger bei der Analyse ihrer Auswirkungen durch die Einbeziehung mehrerer Variablen zu unterstützen.

Abstract

In the current global competitive market, where businesses thrive to reach higher market share by providing sophisticated yet customer-oriented solutions, management and protection of product information along different stages of its lifecycle (from collection of customer requirements to prototyping and launch), is of significant importance. The discipline that provides assurance for product information management is Configuration Management and depending on the scope and complexity of the application environment, requires several activities and measures to be implemented in a proper manner.

This thesis aims at providing an optimization framework for industries to evaluate their level of competence in this discipline and move towards empowering their processes and consequently gain a competitive advantage among their competitors. This goal was achieved by first identifying the critical success factors of Configuration Management and developing a maturity assessment model that evaluates the baseline maturity of organizations with respect to their configuration management efforts and provides a roadmap for further improvements in them. The developed maturity model was then validated through a cross industry assessment and the results laid the foundation for identifying the current state of maturity in several industries, possibilities for enhancing the usability of the maturity model for evaluation purposes as well as the potential research opportunities for solutions that provide enhancements in process capabilities of the discipline.

The primary potential for further development of the model was identified as the lack of quantitative key performance indicators and it was addressed by identification of relevant KPIs that enable quantitative measurement of organizations' maturity. At the same time, an opportunity was identified for conceptualization of a Decision Support System for the Engineering Change Management process that can utilize historical knowledge of past configuration changes to assist decision makers in their change impact analysis by incorporating several variables.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
AKR	Adaptation Knowledge Requirement
ANN	Artificial Neural Networks
AP	Application Protocols
BBN	Bayesian Belief Networks
BOM	Bill of materials
CAD	Computer Aided Design
CBR	Case Based Reasoning
CCC	Configuration Change Control
CCP	Closest Common Parent
CI	Configuration Items
CM	Configuration management
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CMMM	Configuration Management Maturity Model
CMSA	Change Management Support Approach
COBRA	Common Object Request Broker Architecture
CSF	Critical Success Factor
DFMA	Design for Manufacture and Assembly
DoD	Department of Defense
DOD	Department of Defense
DSM	Design Structure Matrix
DSS	Decision Support System
EC	Engineering change
ECB	Engineering Change Board
ECM	Engineering Change Management
ECN	Engineering Change Notice
ECO	Engineering Change Order
ECP	Engineering Change Proposal
ECR	Engineering Change Request
ERP	Enterprise Resource Planning
FCA	Functional Configuration Audits
FMEA	Failure Mode and Effects Analysis
IAEA	International Atomic Energy Agency
IBIS	Issue Based Information System

ICF	Referred to as Inverse Concept Frequency
IDF	Inverse Document Frequency
ISF	Information Structure Framework
ISO	International Standard Organization
KBDSS	Knowledge-Based Decision Support System
KBS	Knowledge-Based Systems
KDD	Knowledge Discovery in Databases
KIT	Karlsruhe Institute for Technology
KM	Knowledge Management
LHC	Large Hadron Collider
MLP	Multi Layer Perceptron
MMP	Maintenance Management Project
PCA	Physical Configuration Audit
PDM	Product Data Management
PLM	Product Lifecycle Management
PMI	Project Management Institute
PS	Proton Synchotron
PSM	Problem Solving Methods
QFD	Quality Function Deployment
RBR	Rule-Based Reasoning
RDFS	Resource Description Framework Schema
SEI	Software Engineering Institute
SVM	Support Vector Machine

1. Introduction

“Without changing our patterns of thought, we will not be able to solve the problems that we created with our current patterns of thought.”

Albert Einstein

1.1 Motivation & Objective

Nowadays, organizations consistently strive to improve their ongoing business activities in order to preserve and expand their market position. One of the key contributors to such efforts is reducing their lifecycle costs and at the same time maintaining their product quality and provide satisfaction and delight for their customers. This can only happen when the customer requirements are reflected in the initial product design and this information is protected from all chaotic changes in various product lifecycle phases.

Configuration management (CM) is a managerial discipline that aims at providing consistency and accuracy of product information throughout its lifecycle. The primary objective of CM is to ensure that in all product lifecycle phases, product information is up to date and consistently match the physical product representation. In a shorter form, CM ensures that products and facilities, including all the systems, equipment and components, are accurately described in suitable documentation at all times (Rouse & Sage, 2011; IAEA, 2010). Many researchers confirm that successful implementation of CM exerts considerable positive effects on product quality, product lifecycle costs and consequently contributes to the ability of organizations to compete in an environment with growing complexity (Dvir, 1998; IAQG, 2011; de Bruin, et al., 2005). This is mainly the case with complex engineering environments where the activities are performed in long lifecycle projects with rather large scopes and the importance of a discipline to keep the consistency of product information is significantly high. Although the benefits of having an effective CM discipline in place are prominent to all professionals, similar to any other discipline, the level of application varies from organization to organization. It is also often unclear what elements of this discipline as well as what extent of their application is required for successful projects and operations.

This gap is typically filled by research sector in the framework of a thorough maturity assessment model that provides the possibility for organizations to compare and evaluate their performance in a certain field against the standards and realize their so called “maturity” level. In the case of Configuration Management, there have not been many works to elucidate a clear roadmap to evaluate organizations’ maturity in the field, prioritize improvement activities and identify an implementation roadmap. This need escalates in higher levels when organizations intend to compare their know-hows in the field with best practices in their industrial sectors and continuously improve their operational capabilities.

Therefore, the goal of this thesis is to identify the Critical Success Factors (CSF) of configuration management in a way towards developing a maturity assessment framework for organizations to be able to measure their CM application quality and move towards improvement opportunities in this field. At the same time, this research aims at utilizing this framework to identify any opportunities where research could further contribute to the efficiency and effectiveness of CM application and consequently provide state-of-the-art solutions for this purpose.

1.2 Hypothesis & Research Questions

The main hypothesis in this research is that the efficiency and effectiveness of configuration management activities of an organization could be improved by utilizing a maturity assessment framework that can measure the current state of those activities and provide solutions based on the identified gaps. The following research questions are to be answered in a way towards proving this hypothesis.

The first portion of the hypothesis focuses on the ability to measure the performance of an organization with respect to its CM practices. Therefore, the first question that faces this research is the following:

How could the Configuration Management practices of an enterprise be evaluated?

The second portion, focuses on the phase after the evaluation where there is a meaningful set of gaps that could be worked on by implementing standard solutions. However, what is most important in this research is to identify the critical gaps for which there is not yet a standard readily available solution.

What is the critical improvement opportunity in the Configuration Management practices?

As the last phase of the hypothesis focuses on providing solutions for any critical gaps in the field, the last research question highly depends on the results of the two previous questions. As could be seen in the next chapters of this work, after providing answers to the first two research questions, the author identified the gap in the current methodologies that support the decision making procedures in Engineering Change Management activities. Therefore, the final research question is formulated as below:

How could the decision making process in Engineering Change Management be enhanced?

1.3 Research Methodology and Structure

In the course of this work, a combination of qualitative and quantitative research methods were utilized depending on the research phase requirements and the nature of the data collection environment. The overall structure of the research flow could very well be harmonized with the PDSA (mostly known as PDCA) Cycle. The PDSA Cycle is a systematic series of steps for gaining valuable learning and knowledge for the continual improvement of a product or process. Also known as the “Deming Wheel”, or “Deming Cycle”, the concept and application was first introduced to Dr. Deming by his mentor, Walter Shewhart of the famous Bell Laboratories in New York.

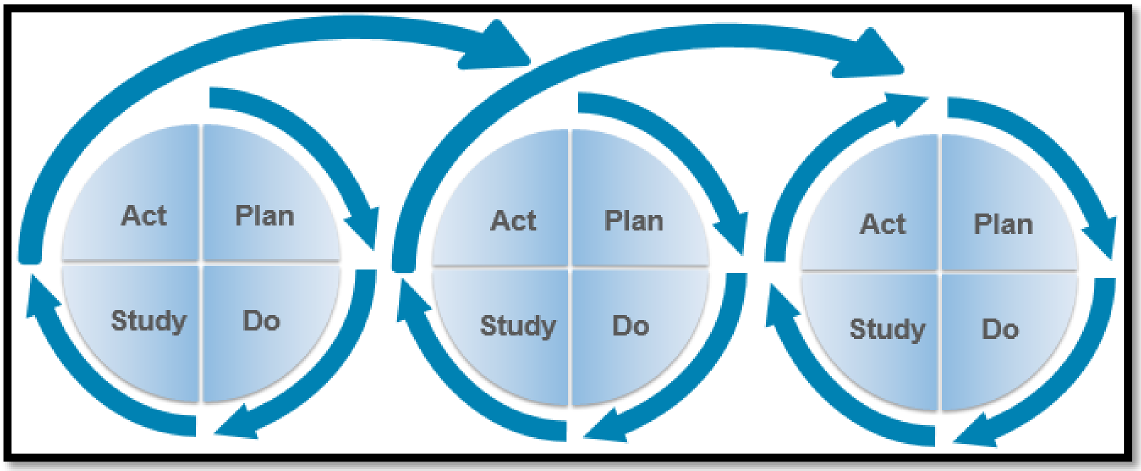

Figure 1 - PDSA Cycle

The cycle begins with the "Plan" step. This involves identifying a goal or purpose, formulating a theory, defining success metrics and putting a plan into action. These activities are followed by the "Do" step, in which the components of the plan are implemented, such as developing the concept. Next comes the Study step, where outcomes are tested to check the validity of the plan for signs of progress and success, or problems and areas for improvement. The Act step closes the cycle, integrating the learning generated by the entire process, which can be used to adjust the concept, change methods or even reformulate a theory altogether. (The W. Edwards Deming Institute®, 2016; Moen, 2006) These four steps are repeated over and over as part of a never-ending cycle of continual improvement. Depending on the scope of the target work, each step could in turn comprise the entire cycle in the way towards completion of the step and moving to next.

The main phases of this thesis, the research methods utilized for each phase and the primary outcome(s) of the phase are illustrated in Figure 2.

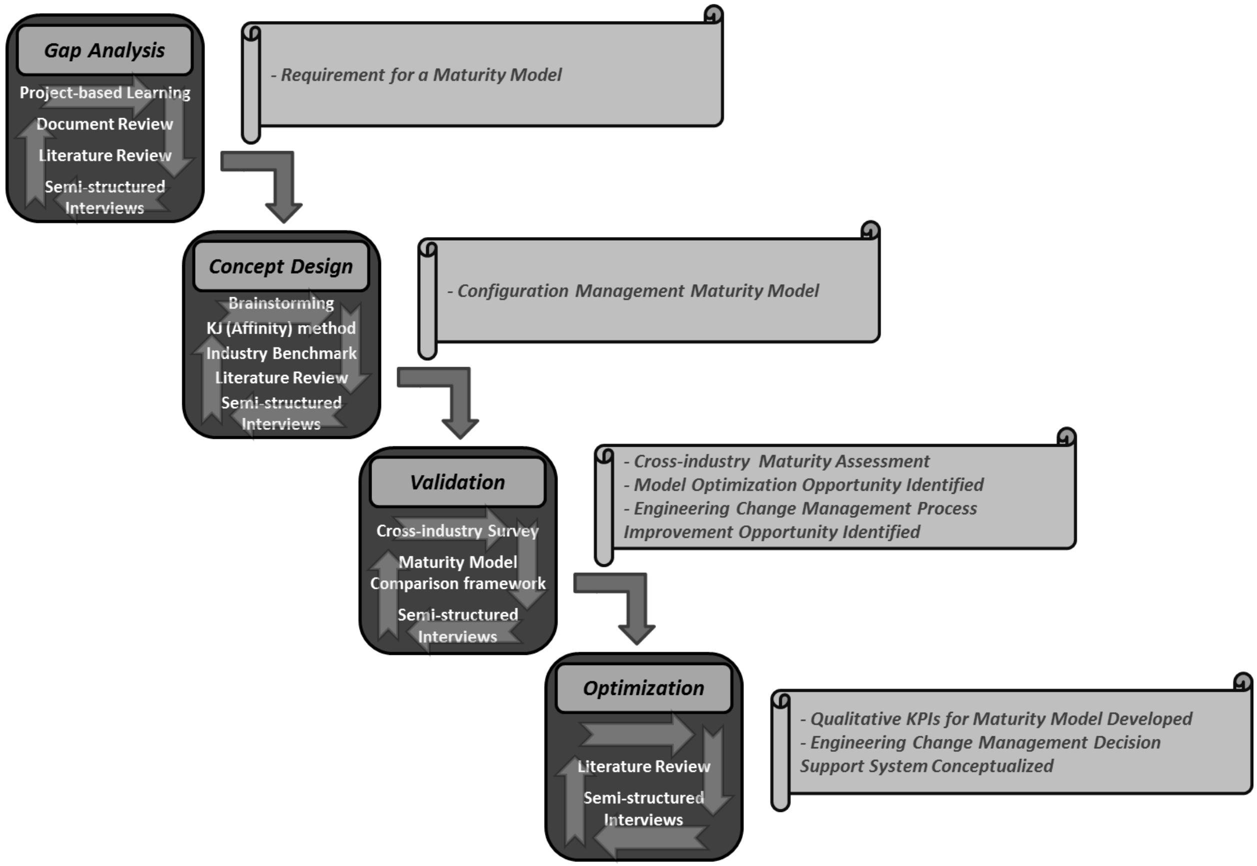

Figure 2 - Research Methodology & Structure

Gap Analysis: In this phase, the primary research subjects such as Configuration Management and Maturity Models are described and an extensive literature review about these subjects are presented. At the same time, to authenticate the realized gap in literature, an ongoing empirical study was also performed. This was achieved by active involvement in the project activities of the target organization as well as internal documentation review and semi-structured interviews.

The outcome of this phase laid the foundation for the concept development in the next phase.

Concept Design: After realization of the gap, a proposed solution was conceptualized in this phase by utilizing KJ method (Scupin, 1997) which is mainly used to structure a set of data gathered from brainstorming or literature review. This led to the characterization of the Configuration Management Maturity Model (CMMM). All components of the model such as capability areas, maturity levels as well as appraisal material were developed and through several rounds of review and brainstorming sessions with subject matter experts, the final version was ready for testing in the next phase.

Validation: The purpose of this phase was to evaluate the functionality of the model and its contribution to finding gaps in research and practice for future process improvement activities. This was accomplished by using two independent parallel actions.

a) A focused research was carried out for creation of a qualitative maturity model comparison framework to act as an independent identifier of improvement opportunities of the model in comparison with several already established as well as novice maturity models.
b) A partnership with industry experts was organized towards implementation of a cross-industry maturity assessment using the developed concept. For this purpose, the developed appraisal material was used in the frame of an online survey that was carried out by utilizing a web-based analytical software. This was mainly chosen due to the benefits of the survey questionnaire (Blair, 2013) with respect to the amount of data collected relative to the time constraint as well as the ease of reaching various geographical locations which was the case for this country-wide research. Several review sessions with industry experts were arranged to provide a well-structured survey to enhance the quality of the gathered information.

The results of the validation phase pointed out several improvement areas both related to the model structure and functionality as well as the research gaps in the area of configuration and change management. This was the basis for optimization activities in the next phase.

Optimization: Realizing the model enhancement opportunities as well as improvement areas in the area of Engineering Change Management (ECM), separate researches were performed to provide solutions for both gaps:

a) A clear set of key performance indicators were developed to represent as an additional detail layer in the model and help the functionality of the model in future organization-specific applications towards clearer understanding of their maturity for each of the CM Critical Success Factor (CSF). This was done via further literature review and interviews with subject matter experts in the industry.
b) A rather extensive research was carried out in the area of ECM to further analyze the gap and propose a conceptual solution for change impact analysis.

Following the PDSA cycle, the results of this research will also act as the basis for future enhancements and process improvement research activities. The structure of the thesis follows the same research phases described above. In the next chapter the initial literature and practice findings are presented. Chapter 3, focuses on explaining the concept development phase. Chapter 4 presents the results of the two validation sections and chapters 5 and 6 each describe one of the enhancement results in the optimization phase.

2. State of the Art in Literature and Practice

2.1 Product Lifecycle Management

2.1.1 Product Lifecycle Management

According to (Saaksvuori, & Immonen, 2008), “Product Lifecycle Management is a systematic, controlled concept for managing and developing products and product related information. The core of PLM is the creation, preservation and storage of information relating to the company’s products and activities in order to ensure the fast, easy and trouble-free finding, refining, distribution and reutilization of the data required for daily operations.”

In one of the widespread definitions by (Stark, 2005), ‘Product Lifecycle Management (PLM) is the activity of managing a company’s products across the complete lifecycle, from the early stages of conception to the final disposal or recycling’, PLM can rather be considered a concept than a system due to its focus on maintaining sustainable market advantage by addressing flexibility and innovation. This concept can be considered as a combination of business rules, processes, methods, and guidelines including descriptions for the practical implementation.

The concept of PLM is very powerful and its benefits include shorter time-to-market, increased innovative ability and profits, fewer engineering changes late in the lifecycle, higher efficiency, and less product faults (Saaksvuori, & Immonen, 2008; Dutta, et al., 2005). Other drivers for the use of PLM consider the demand of more complex products with regards to functionality and components, shorter product lifecycles, customization options because of higher demand standards, management of more complex supply chains, as well as factoring in the increasing regulations concerning safety and environmental issues as noted by (Batenburg, et al., 2006).

An overview of PLM and its systematic concepts with respect to different lifecycle phases is provided by (Ovtcharova, 2012):

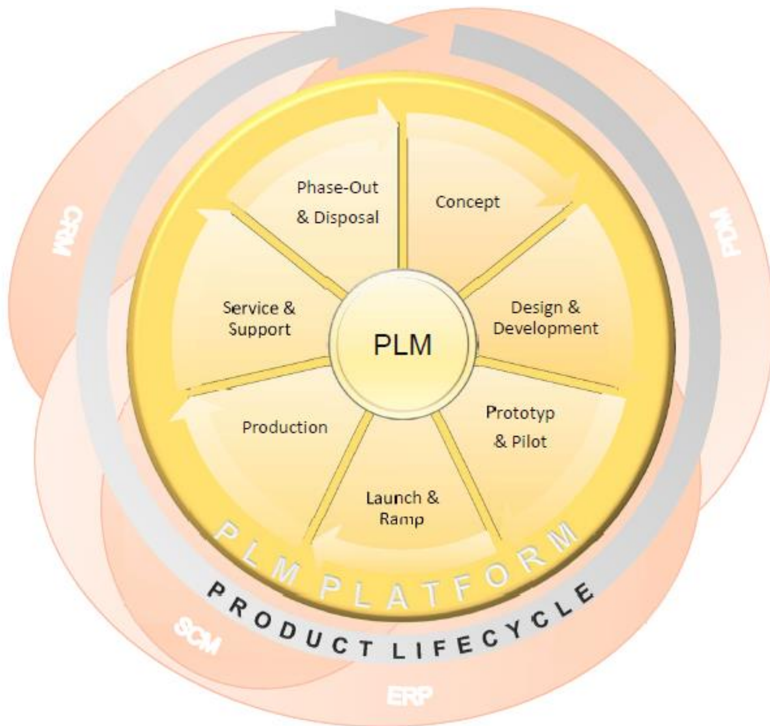

Figure 3 - PLM Overview

Product Data Management (PDM): PDM systems help to administrate and archive all product data such as drafts, product properties, or bills of materials.

Enterprise Resource Planning (ERP): ERP systems are suitable for the planning and organization of production including capacity planning, order processing, administration of bills of materials, and material data.

Supply Chain Management (SCM): SCM systems are responsible for the planning and management of all tasks concerning choice of suppliers, and purchasing, converting, and other tasks of logistics.

Customer Relationship Management (CRM): CRM systems deal with the administration of all customer relationships including address data, history of contacts, offers, or orders. The systems also contain commercial and other customer data which helps to complete an order fulfillment.

2.1.2 System Integration

As described by (Ovtcharova, et al., 2005), adding a system and process integration level to the different information levels in Product Lifecycle Management, fill the information integration gap of proposed solutions and practices in an information structure of an organization. By providing process enhancement methodologies, this layer will integrate the operational processes of an organization to its strategic objectives and ease up the continuous growth of the organization.

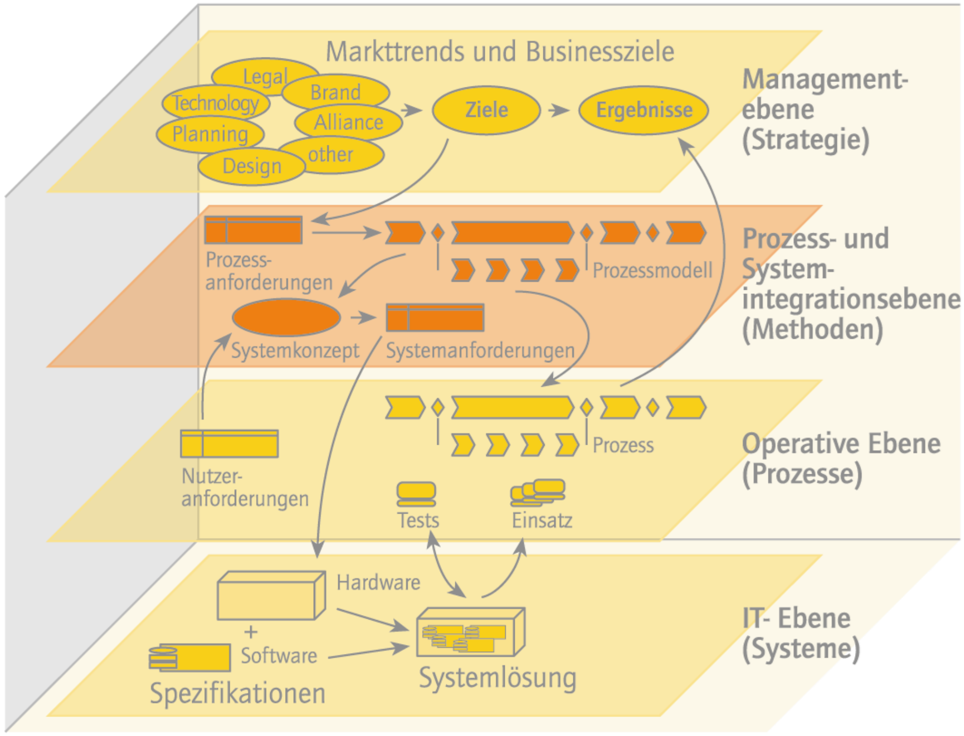

Figure 4 - Information Management Layers (Ovtcharova, et al., 2005)

2.2 Configuration Management

2.2.1 Definitions

Configuration Management (CM) is a discipline that aims at enhancing the consistency and accuracy of product knowledge throughout its lifecycle. As one of the main modules in systems engineering and PLM, this discipline monitors and controls the consistency of the product with its documented requirements at all times (IAQG, 2011; Hass, 2003; IAEA, 2010). CM ensures that products and facilities, including all the systems, equipment and components, are accurately described across all lifecycle phases and all stakeholders are involved in contributing in generating and maintenance of information. It is noted by many researchers and practitioners that implementing effective Configuration Management processes not only improves the efficiency in organizations, but also has direct positive impacts on return on investment, Product Lifecycle costs, on-time deliveries and product quality (Rouse & Sage, 2011).

Configuration Management has been defined by several authors and organizations over time and a few of these are brought below:

"…The Configuration Management is the art of identifying, organizing and controlling modifications to the software being built by a programming team. The goal is to maximize productivity by minimizing mistakes. " (Babich, 1986)

“Configuration Management is unique identification, Controlled storage, change control and status reporting of selected intermediate work products, product components and products during the life of a system.” (Hass, 2003)

“A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status and verify compliance with specified requirements.” (IEEE-STD-610, 1990).

“Process established to maintain the integrity of the work products of a program throughout the system lifecycle to include all system components. “ (Rouse & Sage, 2011)

“The configuration management is a management activity that applies technical and administrative direction over the lifecycle of a product, its configuration items, and related product configuration information.” (ISO/IS 10007, 2003)

2.2.2 History

Configuration management has its roots in the early 1950s when the U.S. Department of Defense recognized the need to improve product documentation during the development phase of its hardware programs in order to avoid additional or duplication of unnecessary components (Thompson, 1997). A few years later, with the NASA Apollo Space Program, a very first concept of configuration management appeared in literature (Fowler, 1993), applied to a complex system. The concept of Engineering Change was then realized with the first Engineering Change Proposal (ECP) being created in 1953. Later on CM was expanded to electronics and firmware and the concepts of maintainability and reliability considerations were discussed. From this point on the real importance and benefits of CM was recognized by others and several principles of what today is understood as configuration management evolved. This was followed by several credible organizations including CM in their agenda such as IEEE and Software Engineering Institute (SEI). In 1984, CM was added to ISO standard and has been one of the fundamental disciplines eversince.

2.2.3 Process Definition

Primary steps and functions of a configuration management Process is illustrated and explained below:

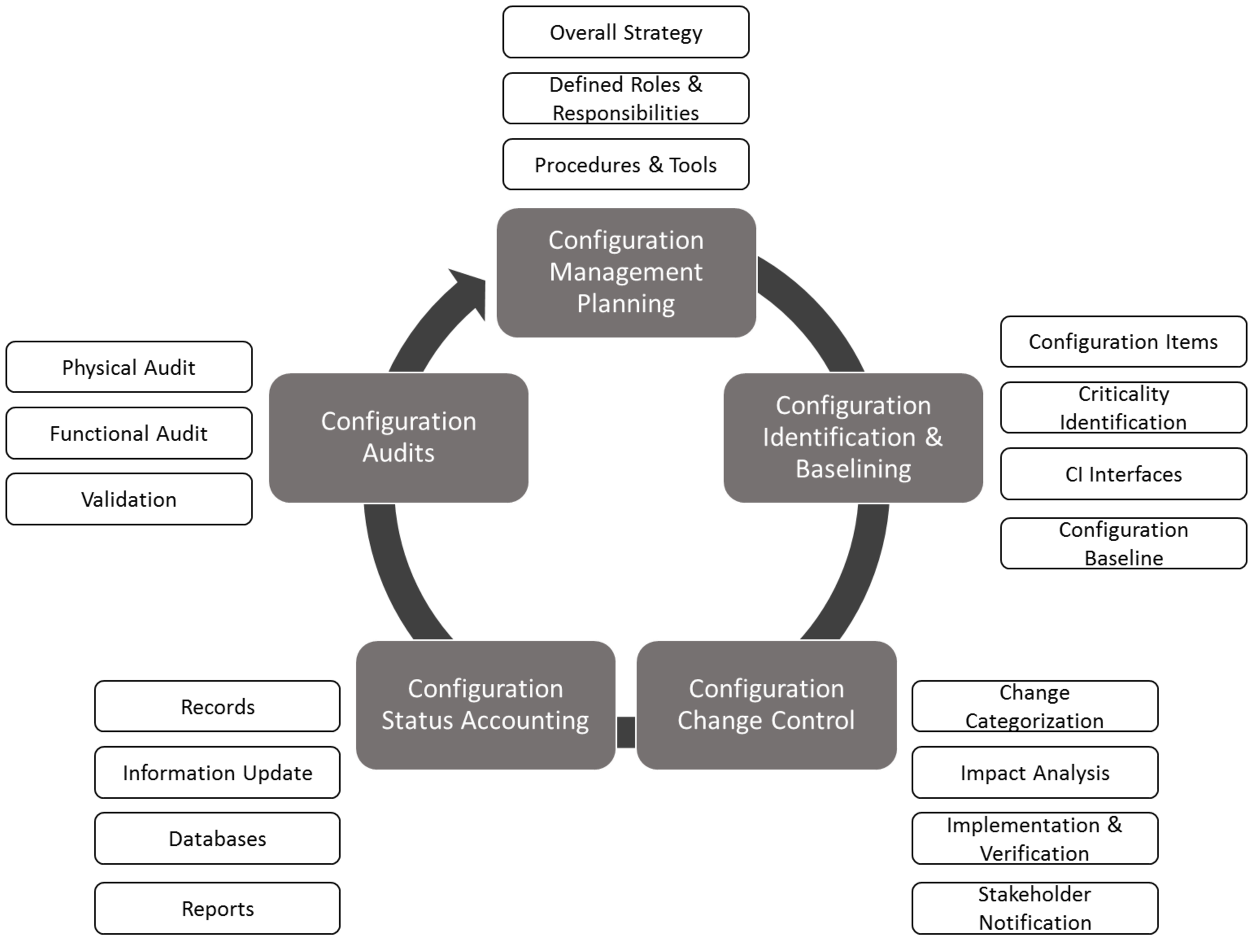

Figure 5 - Configuration Management Cycle

2.2.3.1 CM Planning:

In this initial step of configuration management, the overall structure and plan of activities are illustrated. CM plan has the purpose of establishing the strategic objectives, policies and primary procedures for the implementation of CM discipline. Guidelines as to what software tool or change control process shall be utilized, how the CM organization shall be structures, what types of cross-functional collaborations are to be ensured and decision making responsibilities and procedure frequency are explained. As a vital document, CM plan is to generated and maintained by senior management and updates in the project or organization's CM program is reflected in this master document. (MIL-HDBK-61 , 1997; ANSI/EIA-649, 2011; Hass, 2003; ISO/IS 10007, 2003)

2.2.3.2 Configuration Identification & Baselining:

Configuration Identification is the primary first step in the practical implementation of configuration management. It's the basis of all further activities during the product lifecycle which could impact the customer. This is the step where certain Configuration Items (CI) are defined with respect to their criticality and importance (direct customer requirements, critical assets, high supply chain or quality risk, etc). For the same reason, Cis could vary widely in their hierarchical level in product structure, size, shape, functionality, etc. The first structure and configuration of those items are captured as the first baseline. Configuration Baselines are the latest snapshots of the status of those CIs and their interplay in any of the lifecycle phases. In a successful implementation of CM discipline, Configuration Baseline always matches the physical reality and represent the accurate information about the product. (Lyon, 2000; Hass, 2003; ANSI/EIA-649, 2011; Samaras, et al., 1988; Watts, 2008; ISO/IS 10007, 2003)

2.2.3.3 Configuration Change Control:

Configuration Change Control (CCC), refers to the formal processes involved in identifying, characterizing, evaluating for impacts and approving any changes that targets the form, fit or function of configuration items. This process is in place to make sure the right changes are introduced to the pipeline and correct information about all aspects and impacts of those changes to configuration items or their interactions are captured before approval. In addition to that, CCC ensures that planning for the implementation of changes are done and all change stakeholders are notified about the change in the right time. Once the change was implemented, the configuration baseline is updates accordingly to guarantee the consistency with reality. Configuration Change Control in Engineering Environments are referred to as Engineering Change Management and extensively covered in the next chapters of this thesis. (Hass, 2003; ANSI/EIA-649, 2011; MIL-HDBK-61 , 1997; ISO/IS 10007, 2003)

2.2.3.4 Configuration Status Accounting:

The information about all CM activities are collected extensively and included as reports, records and data sheets in the related configuration documentation. This process is called CSA and is one of the main contributors to the availability of project history and right information. CSA is the enabler for complete, accurate and timely information about functional and physical characteristics of CIs. (ANSI/EIA-649, 2011; ISO/IS 10007, 2003; Hass, 2003)

2.2.3.5 Configuration Audits:

Periodical Audits shall be performed throughout the product lifecycle to verify and validate the physical and functional characteristics of the product with the documentation. After each lifecycle phase when the audits are performed and the latest baselines are validated, the baseline is released to act as the foundation for the next phase. There are two types of audits: Physical Configuration Audit (PCA) and Functional Configuration Audits (FCA). As their name implies each audit type targets certain characteristics of the product for validation. (ANSI/EIA-649, 2011; Hass, 2003; MIL-HDBK-61 , 1997; Samaras, et al., 1988)

2.2.4 Application Overview and Summary

The typical process dynamic of CM including the organization roles and activities involved in each lifecycle step of an engineering project is presented in an abstract level in the following process map. Depending on the size of the organization and scope of CM activities, the following process requires different levels of process steps and details.

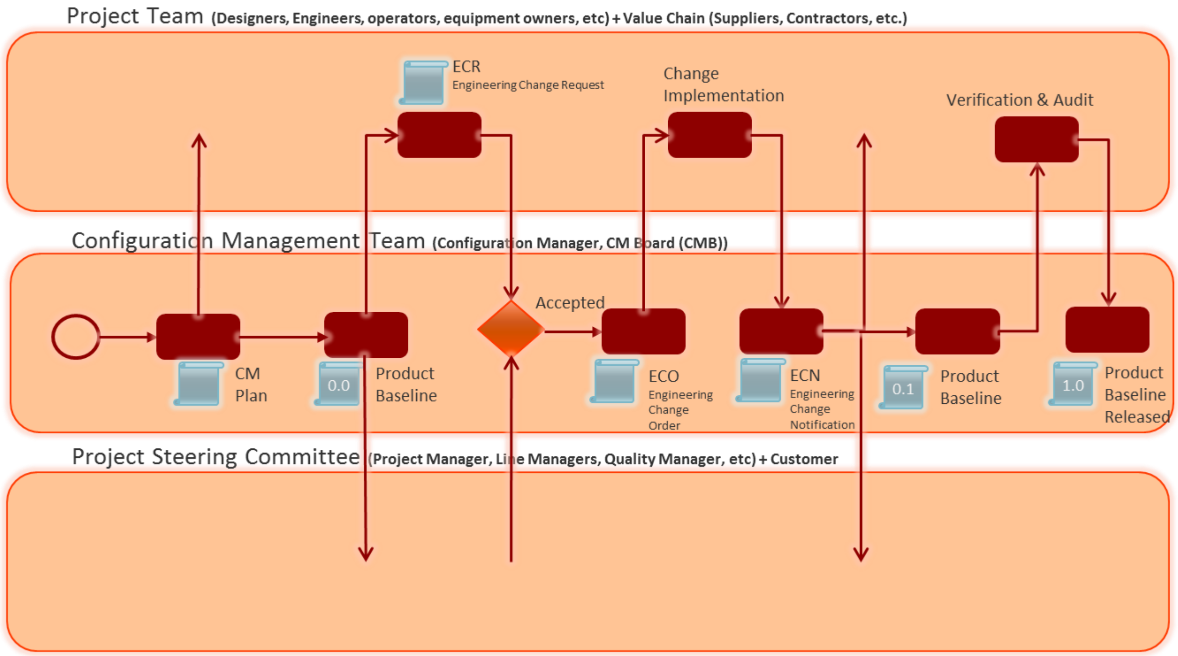

Figure 6 - Configuration Management Dynamics

2.3 Maturity Models

2.3.1 What is a maturity model

Process improvement in any discipline requires the organization to first be positioned with respect to its capabilities and the quality of its processes. It is important for an organization to be able to measure it's as-is situation and identify how mature its processes in different disciplines are. Therefore there is always a need in competitive organizations for supportive tools to assess the as-is situation, prioritize improvement measures and control the progress of such improvements. Maturity models are the essential tools to facilitate these requirements. (de Bruin, et al., 2005; Becker, 2008)

A maturity model consists of a sequence of maturity levels ranging from the very basic level to the profoundly mature level for each important criterion within the discipline being measured. By the use of different investigation tools such as documentation reviews, detailed assessment questionnaires and interviews, the evaluated organization could be analyzed and positioned in the model. Furthermore, by investigating the weakness points and incorporating comprehensive knowledge about the discipline in the model, an improvement scenario could be prescribed with a detailed implementation roadmap with respect to organization's strategic direction and goals. In this way, there will be a vision for the organization to reach this to-be situation step by step and efficiently. (Röglinger, et al., 2012; Saco, 2008)

An example of a maturity model overview could be seen below.

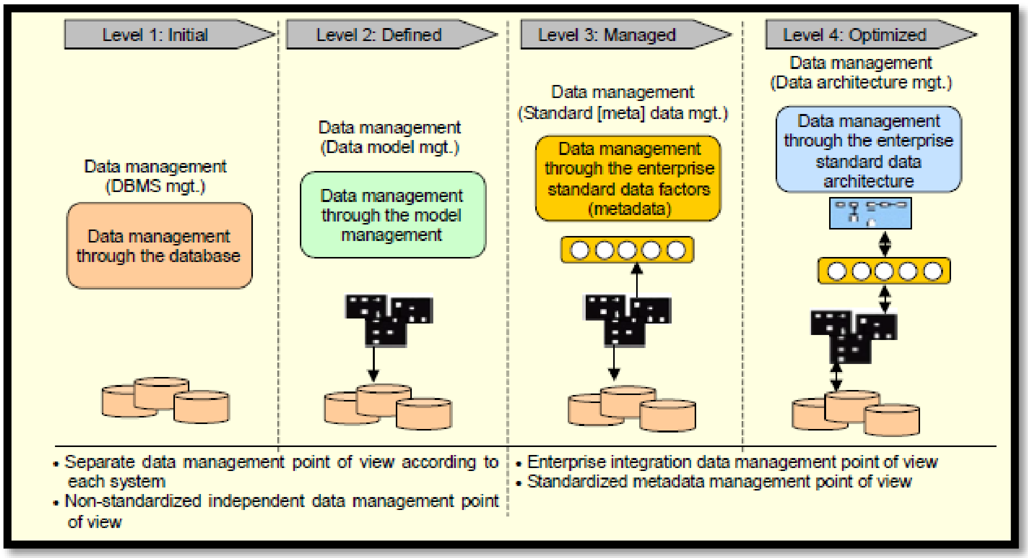

Figure 7 - Data Quality Management Maturity Model (Ryu, et al., 2006)

As mentioned above, maturity models provide excellent support in assessing the maturity of an entire organization with the aim of improving the competitive advantage on dynamic markets. Starting off in 1987, the Software Engineering Institute (SEI) introduced the Capability Maturity Model (CMM), one of the first models to assess the maturity of processes and became the predecessor of many-to-follow maturity models (Software Engineering Institute (SEI) , 2014).

According to (Mettler, 2009), the intention behind maturity models is to provide guidance and direction through an evolutionary process by incorporating formality into the improvement activities. The primary components of every maturity model are (Mettler, et al., 2010; Fraser, et al., 2002):

- A number of maturity levels,
- A descriptor for each level which differentiate it from the previous levels e.g. initial level or optimizing level,
- A generic summary of the characteristics of each level and organization's requirements for reaching that level,
- A number of dimensions which are basically the most important criteria in the discipline being assessed,
- A number of elements or activities under each dimension which are a more detailed and operational requirements for the organizations to reach maturity,
- A description of each element and the way they shall be performed in each maturity level.

These components have to be created based on the benchmark results and best practices in the field on the one hand and on the other hand a thorough literature research including other maturity models in the field, standards and state-of-the-art knowledge.

2.3.2 Developing a Maturity Model

One of the only well-known formal frameworks for creating maturity models is proposed by (de Bruin, et al., 2005). As illustrated in Figure 8 - Development Phases of a Maturity Model, this model recommends six sequential steps in order to create a suitable maturity model for the intended use. Starting from realizing the scope of the problem and the discipline to be assessed, the designer shall then focus on the overall requirements of the maturity model with respect to the

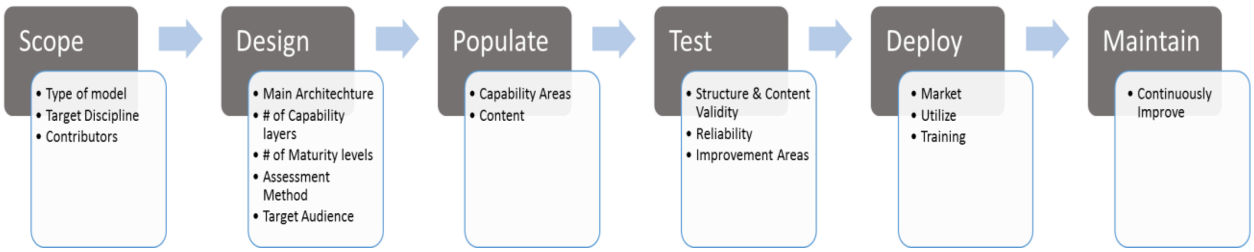

Figure 8 - Development Phases of a Maturity Model

Scope:

As is the case with any concept development, the first point to clarify is the development team and the scope of the content. In this phase the developer tries to introduce the stakeholders in the project and its scope as to what discipline or disciplines are to be covered and assessed by the maturity model.

Design:

At this step, the developer identifies the main overall characteristics of the model such as the target customer, method of assessment, target respondents and the scope of the evaluated organization. Next step in the design phase in to create the main frame of the concept. In this step the developer clarifies the suitable structure of capability areas by establishing the number of granularity layers and also provide the maturity levels and progression method.

Populate:

In this phase, content is extracted and added to the general structure and the domain capability areas are formed by adding components and sub-components. Utilizing literature and empirical studies from experts are the most common methods for creation and validation of the model content.

Test:

After development, the model shall be tested for both the structure and its instruments to assess the target domain. This step provides validity, reliability and generalizability components to the model. Structure validity is whether or not the model matches the target usage and its methodologies are effective in providing the intended information. The content validity is whether or not the domain is fully covered in the model and the assessment results could represent the complete maturity of the target organization in that domain.

Deploy:

Once tested, the model shall be made available for use and generalization. This could be compared with marketing actions for a product and therefore, availability of information about the model and its ease of access shall be considered.

Maintain:

Similar to any other product, maturity models also need to be subject to continuous improvement activities for including up-to-date knowledge and enhancing usability.

Since one of the main intentions of this thesis is to develop a Configuration Management Maturity Model, this model is chosen to structure design activities and the various steps are illustrated throughout the thesis.

2.4 CM Maturity

2.4.1 Configuration Management in Maturity Models

Configuration Management as one of the key elements in Systems Engineering and Product Lifecycle Management processes has always been subject to research and improvements. As explained before, the idea of Maturity Models and capability frameworks has recently become one of the primary approaches towards realizing the organization status and managing the necessary changes in a more efficient and effective step-by-step manner. However, so far Configuration Management has mostly been considered as part of other subject areas and therefore has not been covered comprehensively and with sufficient level of details in maturity models. The available Maturity Models in the field of business processes are mostly focusing on the development in general and although some of them include requirements for CM, the requirements vary from model to model (Hass, 2003). As a result, in this section, an overiew of maturity models which are widely acknowledged, commonly used and more importantly include some senses of Configuration Management in their assessment criteria are presented.

2.4.1.1 CMMI™

Developed by Software Engineering Institute (SEI) at Cornegie Mellon University, Capability Maturity Model Integrated or CMMI™, is one of the most comprehensive and widely used maturity models that covers and evaluates different process aspects of software development organizations. Configuration Management processes are included in this model as one of the sixteen core functions contributing to an organizations software development maturity (CMMI, 2010) and this model is mainly developed for organizations pursuing enterprise-wide process development. The focus of CMMI™ model is on processes to be in place and to work according to the standard definitions and flow. Here, Configuration Management is a support process which shall be followed along with a few other support processes for the organization to reach the second level of Maturity out of five. In the process perspective however, the CMMI™ Configuration Management process description covers most of the high level standard CM activities and extends to the following areas (CMMI, 2010):

- Identifying the configuration of selected work products that compose baselines at given points in time

- Controlling changes to configuration items
- Building or providing specifications to build work products from the configuration management system
- Maintaining the integrity of baselines
- Providing accurate status and current configuration data to developers, end users, and customers

The concept of capability in CMMI™ is defined as the completeness level of each process alone and has three levels. Then based on various capability levels of the organization in various processes, its maturity is defined. As an example, if an organization is in capability level 2 for configuration management and reaches the same capability level for the other six modules that define “target profile 2”, then its maturity level is defined as level 2. If an organization would like to reach maturity level of 3 however, all seven basic modules shall have a capability level of 3 and the next 11 modules shall also reach the third capability level for the organization to be in the third maturity level. The following figure illustrates different levels of capability and maturity and their requirements in CMMI™.

Name	Abbr.	ML	CL1	CL2	CL3
Configuration Management	CM	2	Target Profile 2		
Measurement and Analysis	MA	2			
Project Monitoring and Control	PMC	2			
Project Planning	PP	2			
Process and Product Quality Assurance	PPQA	2			
Requirements Management	REQM	2			
Supplier Agreement Management	SAM	2			
Decision Analysis and Resolution	DAR	3	Target Profile 3		
Integrated Project Management	IPM	3			
Organizational Process Definition	OPD	3			
Organizational Process Focus	OPF	3			
Organizational Training	OT	3			
Product Integration	PI	3			
Requirements Development	RD	3			
Risk Management	RSKM	3			
Technical Solution	TS	3			
Validation	VAL	3			
Verification	VER	3			
Organizational Process Performance	OPP	4	Target Profile 4		
Quantitative Project Management	QPM	4			
Causal Analysis and Resolution	CAR	5	Target Profile 5		
Organizational Performance Management	OPM	5			

Figure 9 - Capability and Maturity Levels in CMMI

Eventhough, configuration management is covered as one of the basic support disciplines without which, organization’s maturity could not improve, the detailed focus on this discipline is still missing and comprehensive dimensions for an individual maturity assessment of this discipline is not completely covered.

2.4.1.2 IAEA

International Atomic Energy Agency (2010), defines the main criteria for assessing the Configuration Management discipline in safety-critical environments based on experience, best practices gathered from different nuclear power plants as well as extraction of IAEA standards. This main set of criteria is briefly explained below.

- Program Management, which focuses on the management support in defining CM strategic objectives and policy, physical scope of CM program, configuration item extraction as well as roles and responsibilities involved in CM activities. Establishing clear terminology and knowledge source as well as identifying the core methodologies to be used in different stages of CM program is of great importance. Furthermore, the information system that enables the organization to perform all CM activities shall be identified.
- Design requirements, which include establishment of formal design requirements for configuration items as well as identification of equipment list and product structure to be used for CM of configuration items (CI). The CIs shall be classified based on the most important design requirements and accordingly the degree of CM for each CI will be identified. New or revised requirements are identified by the authorities based on the organization's strategic mission, lifecycle phase and other relevant factors.
- Information control and change control which include the configuration management process and activities to be performed on information as well as physical entities for ensuring the match between the real facility in place and the available information representing the reality.
- Assessment which focuses on evaluation of organization's activities and processes to identify the potentials for improvement.
- Training which comprises utilizing different means of education for personnel and stakeholders of CM program to extent the knowledge and competence resources in the enterprise.

IAEA, provides an audit checklist rather than a maturity model where the requirements for CM implementation are listed regardless of importance of priority. Also, maturity levels or quality of implementation for each of the practices are not captured. (IAEA, 2010)

2.4.1.3 SPICE – BOOTSTRAP

BOOTSTRAP is a European assessment process which was developed in the 1990s for assessing the capability of European Software industry. It was created based on the SEI's assessment model and ISO 9001 standard. In the newer versions, it took benefit from the ISO standard for Process Assessment (ISO/IEC TS 15504, 2011) as an overall framework for developing maturity assessment models. The goals of BOOTSTRAP methodology are to assess the software development process performed in a software production unit and to provide guidelines for improvement. Similar to SEI Software assessment model (Humphrey, 1987), BOOTSTRAP utilized both Capability and Maturity levels. Although somewhat different in the content, the evaluation and rating process is more or less the same as Capability Maturity Model (CMM) model which is the first version of the discussed CMMI model. After rating the organizations exact overall processes (Maturity Level) and recognizing the strength and weaknesses in the form of attribute profiles

(Capability levels), the methodology offers improvement guidelines that describe how to select gradual changes for improving software production process.

The model utilized two separate questionnaires for data collection before the actual on-site interviews. One set of questionnaire targets the functional organization and the other targets the project organization. The information then becomes complete by supplementing the findings with the interviews, discussions and document evaluations. The BOOTSTRAP process architecture including core assessment areas is shown below.

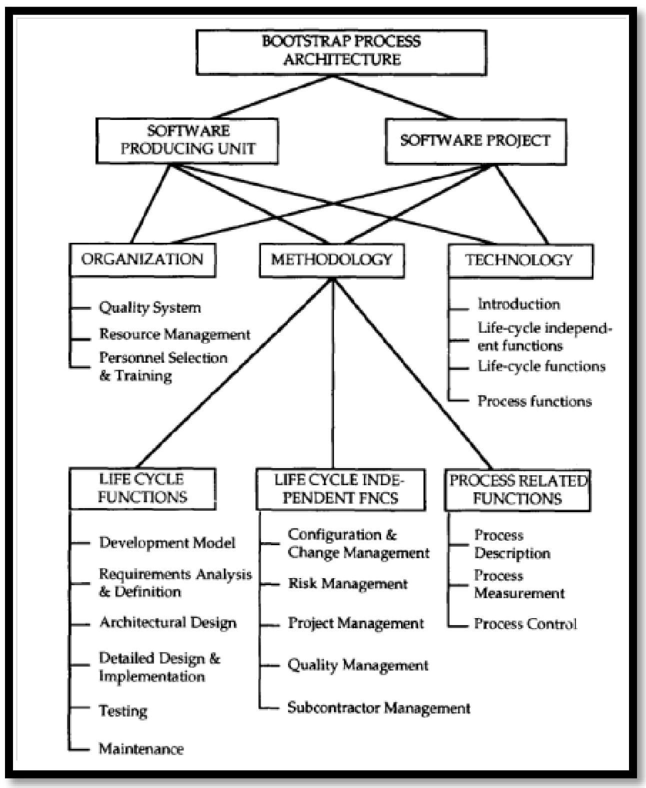

Figure 10 - The Bootstrap Process Architechrute (Kuvaja & Bicego, 1994)

According to (Hass, 2003), BOOTSTRAP 3.2 is a practical implementation of SPICE standard, and they both have almost the same approach and content for assessing software processes. Implementation of Configuration Management, as a support process, is a necessity for process areas to reach capability level of two. SPICE defines CM purpose as:

The purpose of the configuration Management process is to establish and maintain the integrity of all the work products of a process or project.

This is almost an identical definition to CMMI model. However, SPICE does not define goals but states them as a result of successful implementation of the process. For example for CM the following implementation criteria shall be fulfilled:

- A CM strategy will be developed
- All items generated by process or project will be identified, defined and baselined
- Modifications and releases of the items will be controlled
- The status of the items and modification requests will be recorded and reported

- The completeness and consistency of the items will be ensured
- Storage, handling and delivery of the items will be controlled

Maturity of CM according to SPICE model could be defined in the following stages:

To obtain level 1, CM must be performed in such a way that the goals are fulfilled.

In level 2 CM shall be planned and followed up on performance according to the plan. The work products shall be controlled with regards to both quality and integrity. This means that CM process shall be performed on the work products which are the results of the CM process itself. This is the stage where CM is a requirement for all process areas including CM itself to reach level 2 of maturity.

For level 3, the CM process shall be documented in a standardized way. The necessary resources for performing the CM (human resources, tools and equipment) shall be identified and made available.

Level 4 requires that the CM performance and its results be controlled through measurements and if out of control, the process be adjusted.

At level 5 continuous improvement of CM approach is assured by using the process measurements.

2.4.1.4 Project Management Maturity Model

The Project Management Maturity Model is developed by (Crawford, 2007), based on the nine knowledge areas introduced in PMBOK® Guide which is published by Project Management Institute (PMI). This process is also patterned in a similar format as SEI's Capability Maturity Model (CMM).

Unlike CMM™ or CMMI™, the Configuration and Change Management process is not a general criterion for reaching a specific level of maturity for all knowledge areas. That is mainly because the purpose of Project Integration Management knowledge area (which contains CM practices) is mainly integrating different project deliverables and documents and therefore Configuration Management as part of Project Integration Management knowledge area is being assessed as an individual function. However, based on the figure below, taken from PMBOK® Guide (PMI, 2010), most of the other knowledge areas that are involved in producing baselines and deliverables are interacting with Project Integration Management knowledge area for their change requests.

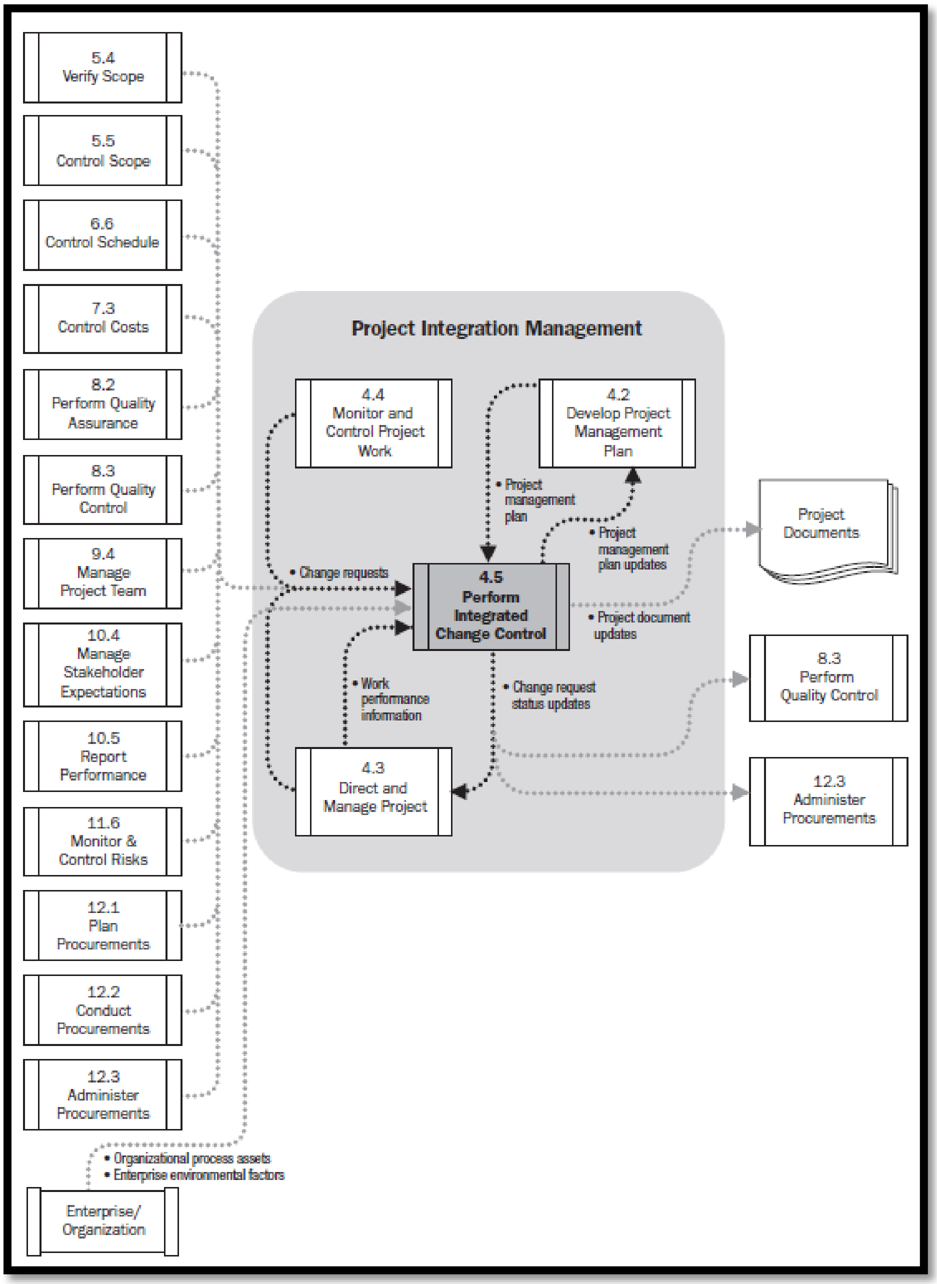

Figure 11 - Project Integration Management Activities (PMI, 2010)

For reaching different maturity levels in Project Integration Knowledge area, an abstract of requirements is brought below:

Level 1 (Initial Process): Communication of the changes to project manager and to the team members is done in an ad hoc manner and there is not a documented change control process in place. Changes are treated uniquely and they are unequally managed and monitored. Configuration Control of deliverables is not in place or poorly managed.

Level 2 (Structured Process & Standards): Scope changes mostly follow a documented change control process. The scope changes are identified for large and highly visible projects and go through a formal change request process to be tracked and approved. There is still not sufficient control for cost and schedule changes since baselining is not performed.

Level 3 (Organizational Standard & Institutionalized Process): There is a defined and documented change control system which provides processes for scope, cost and schedule change controls using change control form and change log. Changes are identifies, assessed, coordinated, managed, communicated to the stakeholders and corrective actions are taken. The process is standard and repeatable and baselines are established, maintained and managed.

Level 4 (Managed Process): All change and configuration management is integrated with the monitoring programs and risk management processes. Functional, physical and data configuration is consistently documented, maintained, managed and controlled for all projects.

Level 5 (optimizing process): There is a full change control process on deliverables which only initiates changes if they are fully understood, documented and approved by management after value proposition. Project changes are included in the determination of efficiency and effectiveness. This also is the case for evaluation of potential changes. There is a certain process for calculating these metrics. There is a continuous improvement process for project integrated change process and perform configuration management activities. Historical changes on projects are examined for finding trends and improving project planning process. (Crawford, 2007)

In this model, Configuration Management is considered in the context of project management and mainly change control function. Eventhough the model as a whole considers the organization structure and other aspects, it lacks a primary and detailed individual focus on configuration management.

2.4.1.5 Systems Engineering Capability Model (SECM)

This American standard (EIA-731.1) is the year 2002 version of the first developed standard with the purpose of enabling organizations to improve capability of their systems engineering processes for better quality product, shorter time to market and less cost (GEIA, 2002). Although this resource is a standard, because of its focus on capability assessment it is presented in the maturity model section of this document.

In this model, Configuration Management is one of the focus areas in the “Systems Engineering Management” category and is interrelated with the “Data Management” focus area in the same category. Both focus areas aim at controlling the content, versions, changes and distribution of the data. However CM focus area mostly targets the control of technical aspects of the product and delivered systems while Data Management module only targets the data.

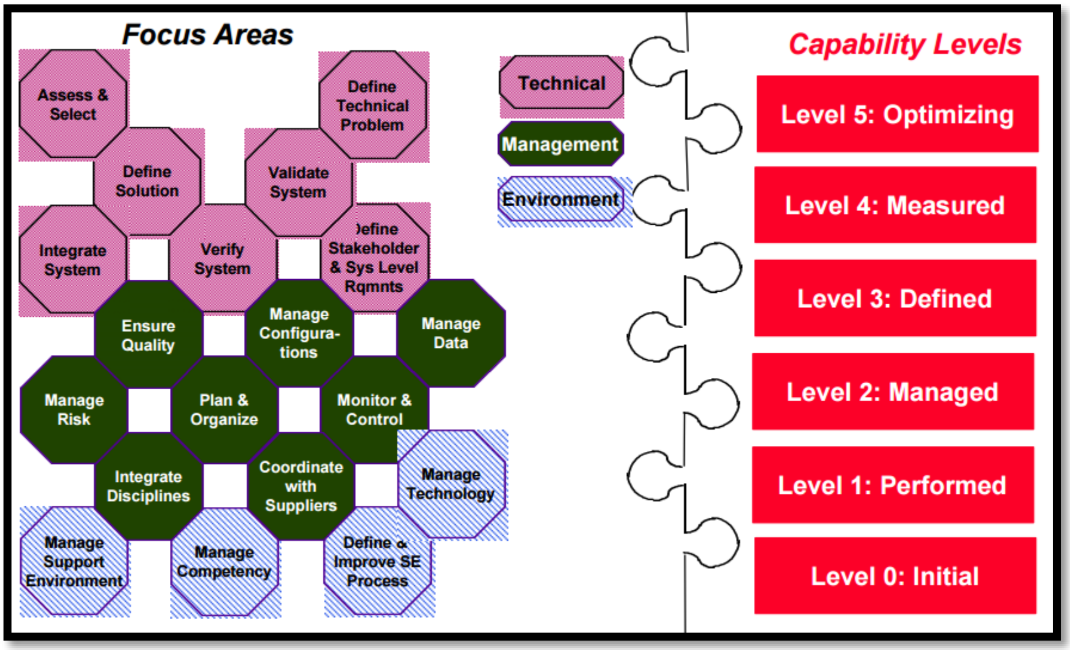

Figure 12 - SECM Capability Levels & Focus Areas

The capability levels are measured based on three main attributes shown in Figure 13.

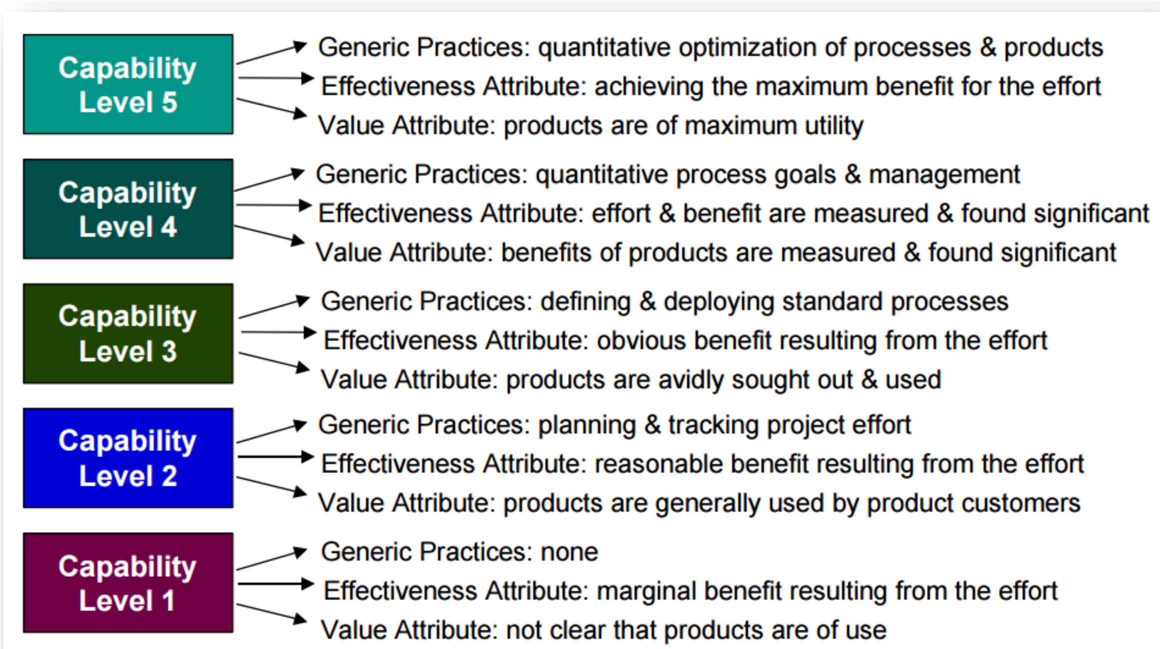

Figure 13 - SECM Capability Assessment Apprach

Based on this standard, CM involves Identification, Change control, Status accounting and auditing of the product and its elements e.g. requirements, interfaces, design representations, etc. The purpose of this focus area is for the customers to have the correct product and the developer to maintain data related to Configuration Items for analyzing and controlling the changes to them. This applies to any type of work product that is defined as a CI and is put under CM including Hardware and Software CIs, design rationale, requirements, product data files and trade studies. The

CM process definitions used in this standard are taken from the National Consensus Standard for Configuration management (ANSI/EIA-649, 2011)

Configuration Identification as the starting point is defined similar to other standards and includes selection and baselining of items which are system, subsystems and documents defining the systems. These will include specification, interface documents and based on criticality in some cases test results. This process step is the act of placing a CI under Configuration Management and managing changes, new versions and releases of the item through a defined process.

Configuration Change Control includes recording, reviewing, approving, controlling and verifying the changes to the established baselines. All changes will go through an evaluation process for a confirmation that the change is in line with the technical and program requirements. Furthermore, the impact of changes to the requirements, contracts and other applications.

Configuration Status Accounting here includes recording and reporting of change information to the baselined CIs. This process step facilitates the traceability of Configuration Identification and the effective implementation of changes.

Finally Configuration Auditing is the act of checking the real item to be in compliance with the baseline and evaluating the accuracy of the baseline documentation. This process validates that the technical requirements are fulfilled by the CI (functional audit) and also the physical product configuration is properly identified (physical audit). (GEIA, 2002)

Eventhough the process characteristics of CM in this standard totally matches the definitions of the CM functionality, several characteristics of CM such as organization, tools, etc, are not comprehensively covered.

2.4.2 Configuration Management in Standards

On the way towards developing a comprehensive maturity model for Configuration Management, In order to capture the essence of Configuration Management and realize the success factors contributing to its performance, there is a need for a review of current mostly used standards. Therefore, the following standards representing some of the widely accepted standards in the field are investigated in this section and their relevant guidelines for CM are briefly explained.

2.4.2.1 ISO 9000-3

ISO 9000-3 is a guideline for using ISO 9001:1994 for software development which although considered as obsolete, is still of interest to many companies. The definition of CM in this standard is brought as following:

Configuration Management provides a mechanism for identifying, controlling and tracking versions of each software item.

The CM system based on this standard shall:

- identify the unique version of each software item
- identify the versions of the sub-items which together constitute a specific version of a complete product
- identify the build status of the software products in each phase of lifecycle
- Control the simultaneous update and modification of each software item by more than one person
- provide multi-location development and update of the items
- Identify and track all actions and changes resulting from a change request from initiation through release.

The next sections of this standard focus on more details about the content of CM plan such as organizational aspects, tools, methods and choice of configuration items along with a description of activities to be performed in different phases of CM process. (Hass, 2003)

2.4.2.2 ISO/IEC 12207 – Systems and Software Engineering – Software Lifecycle Processes

This standard defines Configuration Management and its purpose as (ISO/IEC 12207, 2008):

"The purpose of the Configuration Management Process is to establish and maintain the integrity of all identified outputs of a project or process and make them available to concerned parties"

This standard emphasises on the importance of defining a CM strategy and policy which shall include the authorities for decision making and change control as well as methodologies and storage processes to be used for the CM system.

The CM activities defined in this standard are abstract and are limited to the general steps of planning and execution. In the execution part the organizations, it is suggested to maintain configuration information with an appropriate level of integrity and to ensure that the changes to baselines are properly identified, evaluated, approved, incorporated and verified. For detailed information this standard refers to ISO 10007 which is a specific configuration management standard.

2.4.2.3 ISO 9001

Based on (CMII, 2003), Provisions for Configuration Management in ISO 9001:2000 include activities such as versioning and control of documentation and a more focused set of activities categorized as "Product Realization". The schematic CM related provisions in this standard is shown below.

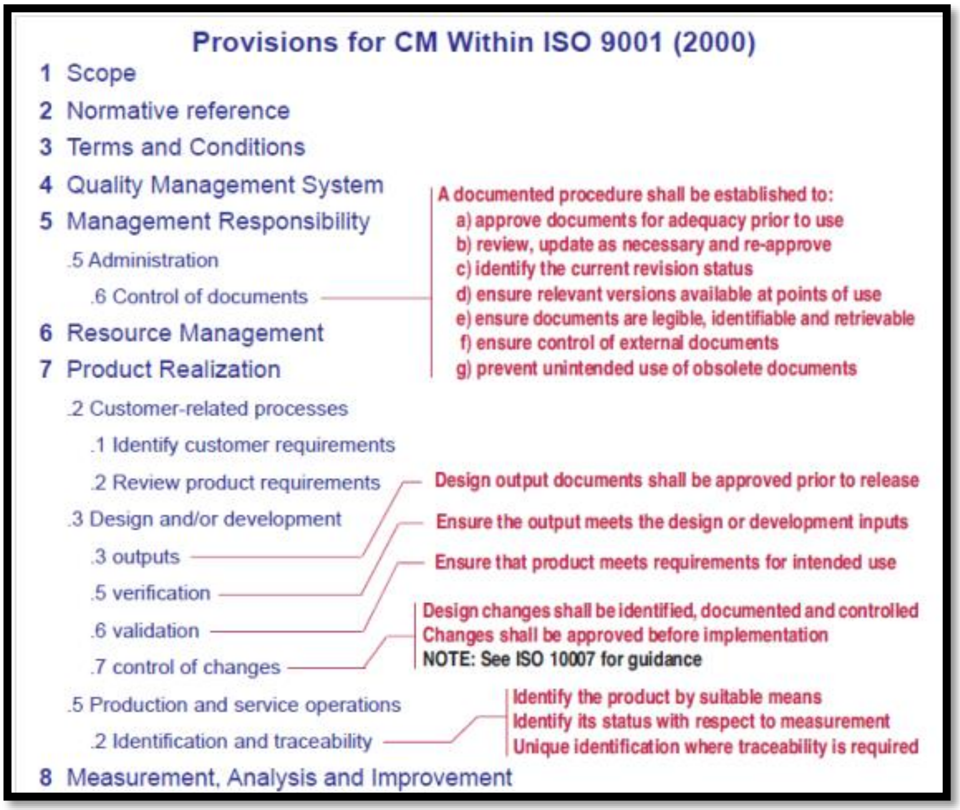

Figure 14 - Provisions for CM within ISO 9001

2.4.2.4 ISO 10007: 2003 Quality management systems —Guidelines For Configuration Management

In this standard, Configuration Management and its functionalities are defined as the following:

Configuration management is a management activity that applies technical and administrative direction over the life cycle of a product, its configuration items, and related product configuration information. Configuration management documents the product's configuration. It provides identification and traceability, the status of achievement of its physical and functional requirements, and access to accurate information in all phases of the life cycle.

This standard is developed to give a better understanding of the subject to organizations and promote the use of CM as well as assist the organization in applying this discipline. Although similar to most standards the information is very abstract in that there is only a brief description of the subject and terminology, responsibility and authority requirements and the process itself, the headlines and overall content covers most of the discipline's extent. Based on (ISO/IS 10007, 2003), CM process is comprised of the main five stages of Planning, Identification, Change Control, Status Accounting and Audit.

Here a bit higher emphasis and more description is specified for Configuration Management Plan and its content. This shows the importance of having a CM strategy and policy as well as a clear set of defined roadmaps and methodologies in addition to responsibilities and authorities to be used in each process step.

2.4.2.5 EIA-649-B – National Consensus Standard for Configuration Management

The US military standard (ANSI/EIA-649, 2011), which replaces the obsolete MIL-STD-973, covers CM principles and practices more comprehensively and defines CM as:

CM facilitates orderly identification of product attributes, and:

• Provides control of product information

• Manages product changes that improve capabilities, correct deficiencies, improve performance, enhance reliability and maintainability, facilitate interface control, or extend product life

• Manages departures from product requirements.

The importance of using a clear set of terminology for CM is acknowledged and followed in this standard. The main functions of CM introduced in this standard are shown in figure 6. These functions match the primary CM processes proposed by most other standards.

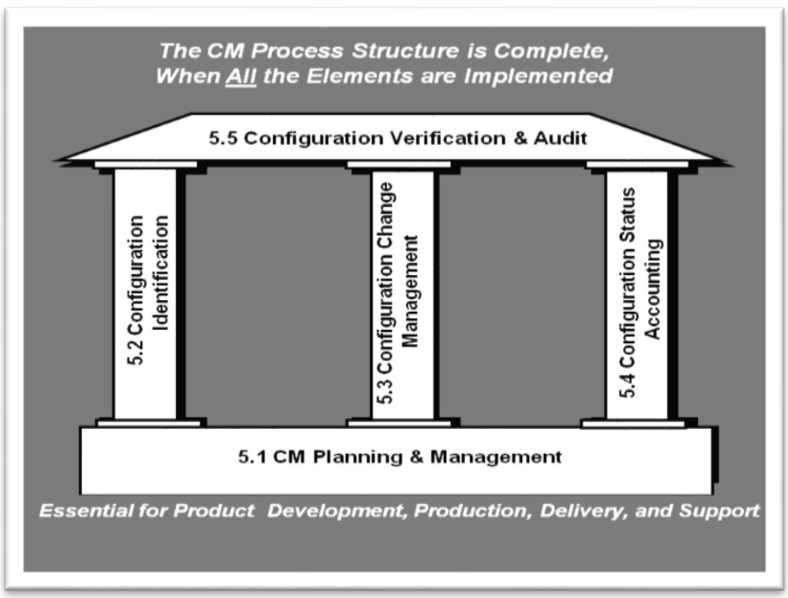

Figure 15 - CM functions in EIA-649-B

Unlike most other standards, the focus on the planning and management of the CM discipline over the product lifecycle is of exceptional interest in this standard. EIA-649-B proposes the following main activities to be followed by organizations in CM planning and management:

- *Implementing policies and procedures, resulting in effective product CM.*
- *Assigning CM functional responsibilities to various organizational elements.*
- *Training of CM personnel and any others who have CM responsibilities.*
- *Determining and applying adequate resources, including CM software tools and facilities.*
- *Establishing CM performance indicators to serve as a basis for continuous improvement.*
- *Ensuring the performance of CM by suppliers.*
- *Integrating the organization's product configuration information processes.*

2.4.2.6 MIL-STD-3046 – Interim Standard Practice for Configuration Management

The US Department of Defense (DoD) is in the process of releasing a new standard for CM (MIL-STD-3046, 2013). The interim version has more or less the same level of comprehensiveness as EIA-649-B, though with more of a focus on the standardization of processes. This purpose is achieved by providing standard process steps, forms and templates for CM functions, such as CM strategy and plan, engineering change templates, functional baseline, allocated baseline, product baseline, configuration status accounting, and functional and physical configuration audits. Comprehensive details for each step of CM process throughout the product lifecycle is thoroughly explained in this standard and additional emphasis of strategy identification as well as CM organizational structure is provided. Wherever needed, clear examples and exceptions are charted to clarify the procedures in practice.

2.4.2.7 Summary

The result of the analysis of CM maturity models and standard are summarized in this section with a table of most important CM characteristics. In the next chapter, these characteristics are extended and combined with empirical findings and form the basis of configuration management capability areas.

	CMMI	IAEA maturity suggestions	BOOTSTRAP	Project Management Maturity Model	Systems Engineering Capability Model	ISO 12207	ISO 10007	ISO 9000-3	ISO 9001	EIA-649-B	MIL-STD-3046
CM Strategy & Policy	●	●	●	●	●	●	●		●	●	●
CM performance measurement	●	●	●	●	●				●		
CM Standard terminology		●					●	●	●	●	●
CM standard processes	●	●	●	●	●	●	●	●	●	●	●
Process support & update	●		●		●		●	●	●	●	●
Process customization					●						●
Knowledge management	●	●		●	●	●				●	●
Benchmarking		●								●	
CM authority &Responsibility	●	●	●	●	●	●	●	●	●	●	●
CM IT support	●	●	●		●		●	●		●	●
Value-chain consideration	●				●			●	●	●	●
Training	●	●	●		●					●	●
Stakeholder involvement				●	●		●		●		●
Management support		●		●	●				●		●

Figure 16 - Extract of Configuration Management Charateristics

2.4.3 CM at CERN

Following de Bruin's maturity model creation methodology, in an effort to understand the intended use, challenges and user requirements for CM Maturity Model in complex engineering environments, and also to observe the application of various CM criteria in practice, a detailed empirical study at European Organization for Nuclear Research, CERN was performed by the author. This was acheived by means of several research approaches. First was utilizing a project-based research approach where the author was involved in an organization wide process improvement project and collected first hand information in the current-state process mapping stage of this project.

Additionally, semi-structured interviews were arranged and performed with key roles in various departments dealing with configuration management. Meanwhile, internal documentations and policies were reviewed.

The general findings and observations regarding the overall CM application, important criteria as well as experienced complications in this subject at CERN is described below and also referred to, throughout the thesis.

2.4.3.1 Scientific Facility Lifecycle

The lifecycle of projects, experiments and equipment at CERN are very long in a way that one generation of employees design and develop the equipment and the next generation supports the operation phase by maintenance, services and finally disposal. Thereforethe consistency of practices in various lifecycle phases as well as departments is a key to smooth management of a project throughout its long life. In a broader perspective, continuous improvement and learning from experience could be of great value and key competency factor for the organization acknowledging the wealth of information and experience in any single project.

2.4.3.2 Facility Complexity

Due to the facility complexity, with an example of Large Hadron Collider (LHC) having more than eight million active assets, the need for a clear status of CM processes, their maturity, their formalization level and range of application at every stage of lifecycle is apparent. This is due to the fact that higher levels of organization and facility complexity automatically enhances the need for more accurate information for any decision making. A few examples of situations justifying the need for more optimization activities in the area of Configuration Management were observed and are described briefly below.

Due to the space constraints in all equipment tunnels at CERN, there are several high-tech equipment available that are hard to reach. The high level of radiation after experiments is another constraint that makes the manual inspection or investigation activities more difficult. In such situation if the facility configuration information does not match the reality, a considerable amount of time and financial resources are required to be spent in re-creation of the baseline at each stage. A simple example is the PS accelerator facility. The Proton Synchotron (PS) is the oldest accelerator at CERN which first accelerated protons in 1959 and has had the role of supplying other accelerators eversince newer generations were built (CERN, 2008). This facility has undergone many modifications over the years to optimize and enhance its functionality. Due to lack of technological solutions and use of hard-copy archiving, the consistency between available facility documentation and the reality is not completely assured. Therefore, resources had to be allocated in a project in 2012 to re-acquire this information and to create the latest baseline again.

A similar example was observed with the cables that transfer the power to the machinery in LHC. Due to the lack of consistent processes during the numerous maintenance and optimization activities that were performed on site, there is a high level of confusion about the map of the currently-in-place cables in the tunnel. This could have high influence on reliability measures, safety aspects and other critical factors of the organization. As it is obvious with respect to the facility complexity and also stated by the personnel in charge, reorganization of the information and matching them with reality in this case needs a great amount of time, budget and human resources.

The root cause to this information loss could come from several situations. One was identified as the emergency maintenance work where modifications are performed in a rush without capturing the information or non-conformity. Lack of comprehensive investigation on the effect of engineering changes before implementation could also be one of the contributing factors. Long project lifecycle and consequently, turnover in human resources with unique facility knowledge could also be blamed for such complexities.

A robust and mature configuration management system that can comprehensively cover such situations, promote cross-functional collaborations with defined roles and responsibilities with respect to CM activities in various phases and incorporate customized processes and tools with respect to the complex requirements of the organization, can very well lower the risk of such incidents and equipment uptime and efficient business operation by injecting first hand knowledge to the project.

2.4.3.3 Regulations

Based on the European regulations related to organizations with ionizing radiations (e.g. The Ionising Radiations Regulations 1999 (Murray, 2012)) equipment information such as their exact previous and current location, radioactivity level, etc., shall be available at all times. In order to satisfy this regulation and also exploiting some benefits such as having more efficient maintenance activities and lower leadtime, this information shall be updated, maintained and available all along lifecycle. At CERN the importance of this traceability requirement is well understood and new initiatives such as Maintenance Management Project (MMP) or Traceability of Radioactive Equipment at CERN (TREC) project (Kepinski, et al., 2013) are considered as priority.

2.4.3.4 Strategic Objectives

The everlasting operational strategic objectives of Scientific Infrastructures such as CERN, is to enhance the safety levels, decrease the human intervention in locations with ionizing radiation and improving the reliability, availability and maintainability. Over time many projects have been performed at CERN with the objective of contributing to these targets. Overall deployment of such objectives with respect to various disciplines has promoted CM as one of the key disciplines at all projects which is a direct enabler for reaching those objectives. A good evidence for the specific acknowledgement of CM as part of strategic objectives of the organization is clearly explained by the description of standard CM requirements in recent quality assurance policies (Mottier, 1999) of CERN. However, in practice the further depoloyment of CM objectives are still not widespread in all projects and performance measures could not be found for success evaluation of CM as an individual discipline.

2.4.3.5 Configuration Management Terminology

Based on the observations and empirical results, the requirement for having a clear set of configuration management related terminology and definitions, is prominent. Considering the complexity of the facility and the interdisciplinary nature of activities (involvement of roles from more than 17 disciplines in each project), using the same vocabulary for addressing different objects and functions is nothing unexpected. The complexity of the issue was realized when the first few weeks of the MMP was spent on providing a clear terminology set and supporting documentation for bringing the same understanding of various concept to all project stakeholders.

2.4.3.6 IT Solutions

Similar to many large organizations, there is a diversity of tools that are used for different purposes including Configuration Management at CERN. Such IT tools are primarily developed in house, over long period of time and by different groups with respect to different project requirements. However, in a discussion with the LHC configuration manager at CERN, it was recognized that the CM practices of CERN are widespread among several tools and there is still no clear understanding of the exact IT tools available in the market, their benefits and comparisons to the developed tools at CERN with respect to CM functionality. Considering this fact, one of the main concerns in each facility is to find or develop a suitable information system that could standardize overall activities of the organization, satisfy the unique scope of different projects and at the same time be customizable and flexible for various requirements.

2.4.3.7 Organizational Roles

Due to the complexity and lifecycle needs of the projects at CERN, several fixed positions in each project (e.g. LHC) are dedicated to CM practices. However, at the same time, several facility modifications are performed by various departments such as maintenance without proper information capture and coordination with CM personnel. An example was observed when on of the equipment were modified due to the maintenance requirements and the small modification blocked the access to a neighbouring equipment. This was realized when a modification was targeted to the blocked equipment and could not be performed which resulted in longer leadtime and several physical configuration audits.

Distribution of organizational roles and responsibilities for Configuration Management processes has a great impact in successful implementation and integration of acticvities with other project modules. The decision to have a fixed organizational section for CM or distributed tasks to project personnel highly associates with the project scope and organization size. Proper organization structure shall be flexible and and promote communication and information transfer in complex environment to avoid such incidents.

2.4.3.8 Role of Contractors

In organizations that routinely work with contractors, the importance of having clear terms and responsibility agreements in the beginning and continuous interactions during the cooperation period is inevitable. Since in most cases the contractors are the ones creating, affecting and modifying the information within the organization, there should be a lucid understanding of how this information shall be captured, recorded and used. As in some cases this could jeopardize the competitive advantages and safety of the enterprise information, there should be a clear authorization process in place to allow the necessary information to be captured from the activities performed by sub-contractors and at the same time protect the organizations information from being compromised in any form. At CERN, contractors and sub-contractors are participating in different activities such as maintenance. As maintenance is one of the core sources of intervention and facility modification, the Configuration Management responsibilities are important to be agreed upon between CERN and contractors. This is the case at CERN to a certain degree. The scheduled maintenance or contractors with clearly defined workorders are rather cooperative and follow a certain agreed process flow. However, whenever there is an emergency maintenance activity that requires fast reaction (sometimes in the middle of the night with no sufficient supervision), the information could not be completely captured. Although most of emergency maintenance is to fix the problem at hand and rarely affects the form, fit or function of the facility, there are certain instances where modifications or patchworks are required to quickly operationalize the facility.

2.4.3.9 Senior Management and Training

In the path to improvement, modifying an organization’s processes and routines does not on its own guarantee the enhancement of CM maturity level. As it’s the case with any change management activities, involvement and support of the senior management as well as availability and accessibility of suitable training shall supplement the change. The continuous training for CM ensures the knowledge transfer among employees and facilitates a communication stream and feedback from people using the CM. The active involvement of senior management in several projects at CERN and their emphasis on frontloading activities and process oriented implementations were observed at CERN. The author noticed a rather huge investment on the initial phase of a project to bring everyone to the same level with regards to the policies and standards to be used throughout the rest of the project lifecycle.

2.4.3.10 Knowledge Management

As described before, CERN and similar complex organizations with long lifecycle projects require a more solid and reliable knowledge management system than others. Moving from people-centered knowledge to robust process design and process-centered knowledge is the key for such organizations to be efficient. Issues with knowledge management were observed and understood by the author in the situations where the information related to equipment or documentations were based on specific personnel knowledge. Several key process documentation were incomplete and confusing where the process owners had left the organization long before. The same applies to the technical knowledge about facility or operation where extra effort had to be put on recreating the lost knowledge.

2.4.3.11 Continuous Improvement

Continuous improvement initiatives for further developing the processes and the knowledge sources shall be integrated to CM processes. Internal and external benchmarks for best practices could help the organization utilize the innovative ideas and cope with emerging problems. CERN, as one of the highly influential research institutes in the world, is filled with experts and scientists from all over the world to share their state-of-the-art knowledge to create unique results. As much as this applies to the technical research at CERN, there is still room for utilizing best practices in various operational disciplines such as configuration management. Utilization of such resources especially from similar organization types could highly impact the maturity of organizations business processes.

3. Configuration Management Maturity Model (CMMM)

When it comes to maturity assessment frameworks, Configuration Management still lacks an individual focus and if ever, it has mostly been considered as a part of other disciplines or maturity assessment frameworks. Consequently, due to the scope limitations of such maturity models, this subject area has not been covered fully and comprehensively.

In order for organizations to have a clear understanding of their full potential on any subject, there is a need to have a wider focus and measure its individual maturity among other business processes in place. Evaluating CM exclusively would give the organizations the opportunity to realize the sole benefits of improving this discipline for resource allocation and investment justifications.

To fulfill this gap and to move one step towards the ultimate goal of this thesis for providing solutions to measure and optimize the overall CM maturity in various organizations, a configuration management maturity model is developed in this section.

Following the development framework by (de Bruin, et al., 2005), the main capability areas or Critical Success Factors (CSF) extracted from literature and practice are presented in the next section. Followed by that, various maturity levels are identified and expectations for organizations in each level is explained. Lastly, the integration of maturity assessment and improvement activities with routine Configuration Management activities of an organization is illustrated in the last section of this chapter.

3.1 Configuration Management Capability Areas

In order to identify the primary capability areas for configuration management, first the information that was gathered from maturity models and standards was condensed to several categories (Figure 16). These categories were then merged due to close relationship and the following five abstract levels were recognized as the primary configuration management dimensions.

Strategy & Performance	The existence and alignment of CM strategy and policy with the overall strategy, vision and mission of the entreprise. A set of performance indicators to measure this alignment and the achievement of CM goals.
Processes	The availability of standard processes that can stabilize the CM activities among all organizanional units throughout product lifecycle and the approach to test and measure the efficiency of such processes.
Organization & Value-stream	The organizational structure of the entreprise with respect to CM and the level up to which the roles and responsibilities are defined. The ability to cope with the complexities created in CM activities throughout the supply chain.
Information Technology	The information systems and tools being used by the entreprise and the level of match between the functionalities and the necessities. The level of integration of these information systems into the business processes of the organization.
Knowledge & Support	The level up to which the entreprise is knowledgable in CM practices and how actively do the senior management support and promote the application and enhancement of CM activities throughout the organization.

Figure 17 - Primary Dimensions of Configuration Management

Summing up the finding from literature and empirical research, various CSFs under each dimension were qualitatively identified (Figure 18). These findings were reviewed with experts from engineering organizations for applicability, understandability and discipline coverage.

Configuration Management

Strategy & Performance	Process	Organization & Value-stream	Information Technology	Knowledge & Support
Vision, Mission & Objectives	*Standard Processes*	*Org. Structure w.r.t. scope*	*Supportive IT System*	*Standard CM Terminology*
Strategy Depoloyment	*Process Integration*	*Well-defined Roles & Responsibilities*	*System Integration*	*Comprehensive Training Program*
Communication of objectives	*Process accountability*	*Enhanced Cross-functional Collaboration*	*System Flexibility*	*CM Resources*
Key Performance Indicators	*Accessibility to stakeholders*	*Lucid Contractual Agreements*	*Visualization & User-friendliness*	*Senior Management Involvement*
Continuous Performance Measurement & Improvement	*Process Customizability*	*Stakeholder Involvement*	*Access Rights Management*	*Communication of Benefits*

Figure 18 - Configuration Management CSFs

3.2 Maturity levels

Since CM is one of the support disciplines in systems engineering and product lifecycle management, one of the key maturity indicators for the success of this discipline was identified as the extent of its application among other disciplines in various organizational levels. The more organizational units and projects incorporate certain activities in their routine processes, the more robust and standard those activities become throughout the organization. This way, over time and through continuous improvement of such ubiquitous routines, organization's maturity in the field enhances to a high extent. This concept of maturity and organizational alignment, was introduced by (Scheper, 2002) and was further validated by (Batenburg, et al., 2006; Batenburg & Versendaal, 2008). Respectively, the four developed maturity levels for CM is illustrated in the following figure and described in more details in the next sections.

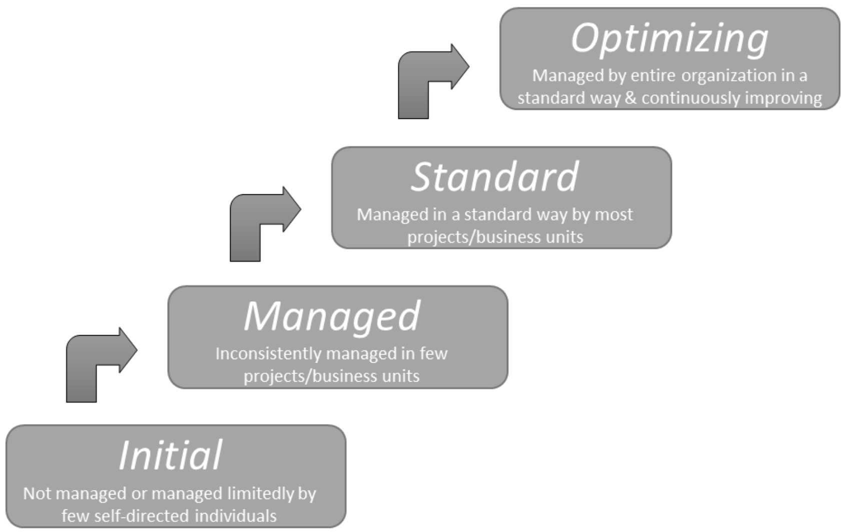

Figure 19 - CM Maturity Levels

3.2.1 Initial

In organizations with the maturity level of “initial”, Configuration Management is not understood as a stand-alone discipline and therefore, it is not included in the strategic path of the organization. Due to low experience in the topic, there is not enough information among personnel about CM and it is rarely promoted in routine activities. There is no terminology set and knowledge source defined for CM and therefore, reasons for failure and inconsistency between reality and documentation is not well realized.

There are no formal processes in place related to CM functions and the engineering changes are approved or rejected with limited rationalization and analysis. The information technology might be used for very basic CM functionalities such as naming, numbering or versioning but the importance of utilizing CM functionalities is often neglected. The level 1 organization is characterized by lack of information about the real product/facility as well as low quality, out of scope, over budget and over schedule projects caused by not practicing formal configuration management processes. Non-conformities among documentation and physical project deliverables are prevalent.

3.2.2 Managed

Organizations in this level do not have a clear Configuration Management strategy and policy in place and teams primarily rely on their own grasp of objectives behind CM application which might not be in line with the project’s or organization's overall strategy. The processes are ad hoc and different among different departments. There is a lack of accountability in managing and enforcing the CM processes and therefore often individuals and their knowledge are the primary source for decision making. Decision about engineering changes are taken by individuals and their overall grasp of situations rather than full understanding of the change impacts.

There is not a standardized set of terminology and definitions for CM in place and therefore different opinions are introduced as standards based on different backgrounds. The organization structure does not appreciate the necessary roles for CM activities and the tasks are divided among different teams as they come up. There is little appreciation and acknowledgment for CM practices among senior management and therefore, formal training in this area is not supported.

Configuration Management requirements has not been considered in the choice of information technology. Therefore, only a few of the needed functionalities are supported in the projects. In best cases the functions are performed by self-developed tools. The maturity level 2 organizations are characterized as having projects which frequently exceed the budget and schedule and deliver lower quality products caused directly by lack of suitable Configuration Management activities.

3.2.3 Standard

At this level due to the well-understood need for CM discipline, clear strategy as well as a CM policy is developed for each individual project. Objectives are lucidly defined and controlled during the project. CM processes are defined in the beginning of each project and are accessible by everyone. These processes are developed and managed based on the knowledge of involved personnel in the project and do not comprise previously used processes and lessons learned from other projects. Configuration changes are managed according to the project processes but could still benefit from cross-functional decision making sessions.

The importance of CM is understood by the personnel in charge of each project and they support the implementation of the discipline. CM terminology and knowledge-base is created and distributed by certain knowledgable individuals among project team members. CM roles and responsibilities are assigned in the beginning of each project and vary depending on the specific opinion, knowledge and experience of project managers in different projects.

Information system, whether developed in-house or chosen from commercially-off-the-shelf products, is used for the CM activities and requirements of each project. There may be similarities among the systems used in different projects but there is not a standard choice or framework for choosing the IT system for the whole organization.

The maturity level 3 organizations are characterized as organizations with projects that have limited problems in meeting schedule, budget, project scope and product quality that would be caused directly by CM processes. However, there is a lack of consistency of the entire organization's CM activities and benefits where some projects benefit from CM in a higher extent than the others.

3.2.4 Optimizing

At this level, there is a specific corporate Configuration Management strategy and policy in place that the entire organization adheres to. This strategy is part of the overall enterprise strategy and is deployed to different levels of organization i.e. strategic, tactical and operational levels, in a consistent and standard manner. There are quantitative measures to evaluate the obtainment of overall CM objectives in a regular basis.

Standard CM processes are defined for the whole organization and are accessible in a central process repository for all stakeholders. These processes have key owners who are accountable and responsible to provide training and to develop the processes further by collecting feedback from allover the organization. There is a clear methodology for engineering change management which involves strong guidelines for cross-functional change impact analysis. CM terminology is distinguished from other disciplines' and standardized through the whole organization. This set of terminology along with CM knowledge base is available for all stakeholders and the correct usability is assured.

A corporate centralized CM organization is identified which supports the work of CM-related personnel in various projects. Clear roles and responsibilities are defined and communicated to the stakeholders. Organized practical CM trainings are provided by the corporate CM department which assures the consistency of applications all along the organization. There is a knowledge management system in place where the learnings from various projects are captured and actively utilized in decision making activities. CM is supported by senior management and continuously promoted through success stories and association of clear KPIs.

When choosing information technology, the CM functionalities and requirements are taken into consideration. The entreprise IT system is capable of incorporating configuration management modules with an acceptable level of customizability. The information is globally available through the whole organization and clear authorization system is in place for creating, using or modifying any information.

The maturity level 4 organizations are characterized by having a consistent performance in all projects. The CM activities are being performed during the entire project lifecycle and main project factors such as schedule, scope, budget and quality are met thanks to the efficient CM system. A standard way of working for CM processes is in place and knowledge is being built and reused continuously. This helps the organization towards constant growth which is tracked and observed using the success measures.

3.3 Business Process Integration

As the concept of maturity assessment in general is developed to "continuously" improve the business processes of an organization with respect to a certain discipline, it shall be integrated into the routine activities of the organization. After mapping the standard Configuration Management business processes of an organization in Figure 6, the following high level process diagram was created to illustrate the integration of Configuration Management Maturity Model to this routine process set, but with a longer term implementation scope and focus. This utilizes the system integration layer proposed by (Ovtcharova, et al., 2005) and illustrated in Figure 4 - Information

Management Layers where the operational excellence layer acts as the layer between organizational strategy and the routine CM processes of the organization.

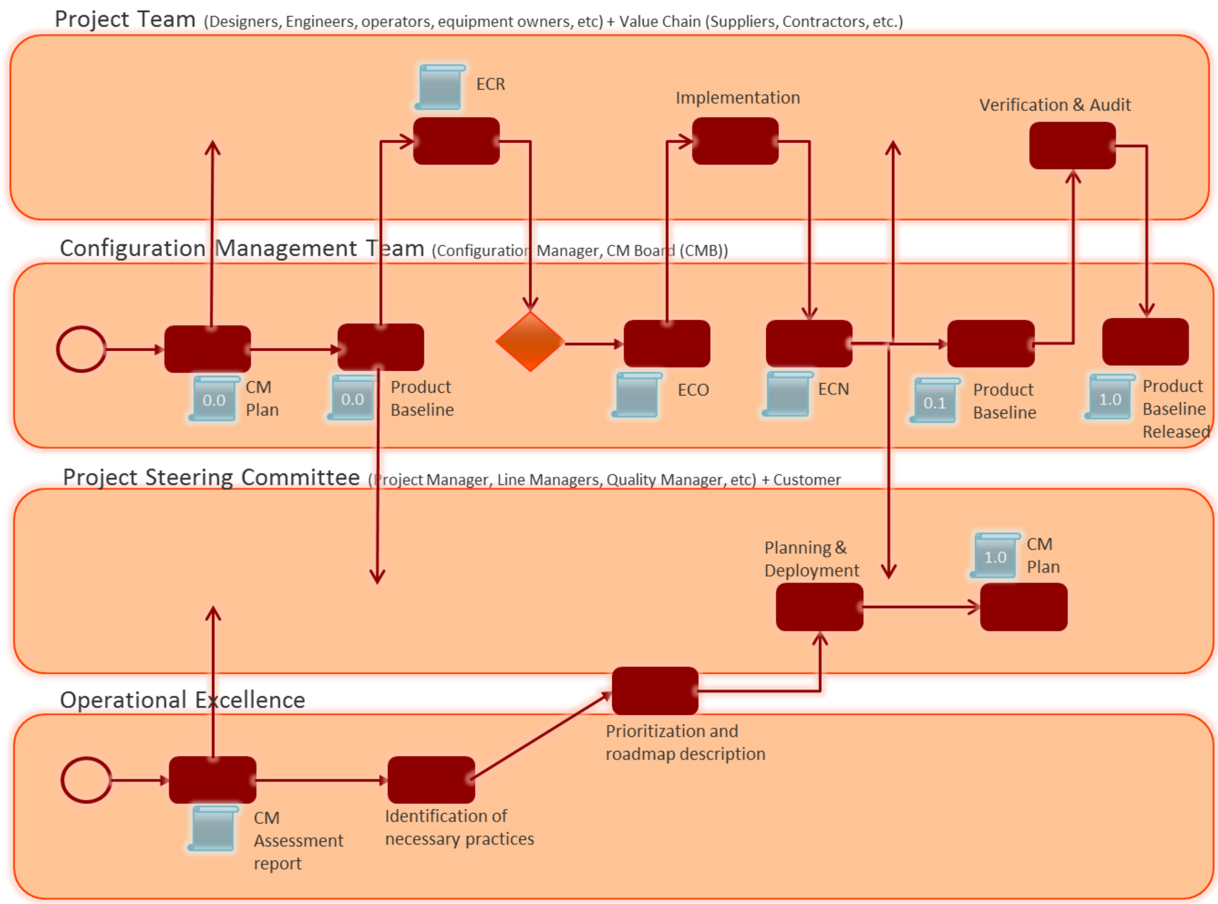

Figure 20 - Integration of CMMM with CM Dynamics in an Organization

Depending on the industry and the lifespan of products and projects in the target environment, the benefits of maturity enhancement activities could be observed either in the same project or after several learning practices in various projects. An example for long lifecycle projects could be the LHC project at CERN or creation and operation of a nuclear power plants where one generation of engineers are involved in the design phase and the next generation is involved in the operation. In such complex environments, there is a great opportunity for self-reflection at each lifecycle phase which if exploited properly can be the basis for continuous improvement as an important function to update the processes and procedures gradually. This guarantees that the CM methodologies, processes and tools are sufficiently mature and match to the world standards at all times and the facility is managed in the most efficient and effective manner.

As can be seen in the figure, the configuration management plan includes the most updated directions, policies and procedures which is the basis of CM activities in the entire lifecycle of the project. In parallel to the routine activities, the maturity status of the organization is regularly assessed against CM Maturity Model by the operational excellence office. The results of the maturity assessment will clarify the main CSFs that require further attention and can identify the next strategic direction and objectives of the CM discipline that shall be included in the next version of the CM plan.

4. Concept Validation and optimization

As described in section 1.3, two methods were used in conjunction to tackle the validation section from two angles. First, in order to validate the nature, structure and usability of the developed maturity model, a maturity model comparison framework was developed. This framework was utilized to position the CMMM concept among other already established maturity models in the field and perform a gap assessment for future developments to model structure. The details associated with development of the comparison framework as well as the results of comaparison are available in section 4.1.

Second, in order to validate the content and coverage of the model for CM discipline as well as creating a CM industry-base benchmark for all organizations, the model was put in use. For this purpose, an extensive appraisal material in the framework of a questionnaire survey was developed on the basis of the CM Maturity Model. In a collaborative project (Niknam M, et al., 2013), the survey was distributed to a wide range of organizations from various industry sectors. The results of this cross-industry maturity assessment provides a better realization of the level of maturity in various sectors and at the same time by targeting subject matter experts in all respondant organizations, evaluates the functionality of the maturity model itself. The extensive results of this study could be seen in section 4.2.

4.1 Maturity Model Comparison Framework

4.1.1 Development of the Comparison Framework

As stated by (Wendler, 2012), there is a lack of research in the area of maturity model validation including a framework for structural comparison and evaluation. Therefore, the focus of this section is to provide a framework where the availabiltiy of various characteristics of maturity models could be checked and compared to successful models in the field.

As there are few recognized works in characterizing the structure of a satisfactory maturity model, a combination of all ideas were used and three main attribute dimensions were utilizied to develop the structure of the comparison framework (Mettler, 2009; Mettler, et al., 2010; de Bruin, et al., 2005; Man, 2007). With the help of research and industry subject matter experts these dimensions were further enhanced in details in order to fill the gaps in covering all aspects of maturity models. The characteristics in each dimension were clearly defined and a scoring system were recognized to be able to rank each model with respect to those characteristics.

After an extensive literature review in the field, several establsihed maturity models in the field of Product Lifecycle Management were identified to be compared with eachother as well as the developed model in this thesis. The structure of the framework as well as the results of the comparison can be found in the following sections.

4.1.2 Framework structure

4.1.2.1 General Attributes Dimension (GAD)

The *general attributes dimension* will cover two blocks of attributes. The first block will include the "basic information", which represents top level information that is necessary to get a simple distinction between all kinds of models. This includes the name of the model, the acronym (if exists), the primary source (where information was collected), the addressed topic, the origin (academic or practitioner-based), the year of publication, the accessibility (free or charged), and the addressed audience (management- or technology-oriented). The second block "structure details" will go more into detail on aspects like the covered business dimensions and maturity levels as well as possible testing parameters.

4.1.2.2 Design Attributes Dimension (DAD)

This dimension will focus on design related issues concerning the maturity models. This includes the main purpose of the model, which can be either *descriptive* (the application is a single point encounter with no intention of maturity improvement or analysing performance relationships), *prescriptive* (analyzing domain relationships in order to boost business performance and thus increase the business value, therefore identifying gaps and creating a road-map for improvement), or *comparative* (performing an industry-wide benchmark across different organizations to compare similar practices) (de Bruin, et al., 2005). As mentioned by Mettler the concept of maturity covers the focus of the model, including *process maturity* (to what extent a specific process is explicitly defined, managed, measured, controlled, and is effective), *object maturity* (to what extent a particular object like a software product, a company report or similar reaches a predefined level of sophistication), and *people capability* (to what extent the workforce is able to enable knowledge creation and enhance proficiency) (Mettler, 2009).

Additionally, (Kärkkäinen, et al., 2012) introduced a fourth *customer* dimension (capability for management of all customer-related data, information and knowledge concerning the whole product lifecycle), which will be factored into the comparison. The composition shows that a model can either be concluded as a *maturity grid* (text descriptions for each activity at each maturity level with moderate complexity), a *Likert-like questionnaire or hybrids* (questions are statements of 'good practice' to score the relative performance; hybrids combine this with a maturity grid), or something altogether different *such as KPIs for each dimension* (e.g. CMMI™). As for the chosen assessment approach, the model can either be *staged* (the model matures the organisation as a whole) or *continuous* (improves capability of specific processes within the organization) (Gupta & Rao, 2011). Whether a model is designed to be *one*- or *multi-dimensional* (being adaptable to multiple domains) will also be covered in the scope. The flexibility or rather adaptability of a model can either be represented as a change in its *form* (e.g. the underlying meta-model or model schema, the descriptions of the maturity levels or question items) or its *function* (e.g. how maturity is assessed) (Saco, 2008). As the last criteria in this dimension, the reliability of the model will be addressed. When there is at least one testing available, it can be assumed that the model has been *verified* but only until a model has been thoroughly tested and accepted by many practitioners it could be considered *validated*.

4.1.2.3 Usage Attributes Dimension (UAD)

The *usage attributes dimension* covers five attributes mainly concerning application issues. First off, the method of application is defined by either being done by a *self-assessment*, *a third-party assisted assessment*, or be concluded by *certified practitioners*. A variety of instruments is used for this application, including *document reviews*, *work groups*, and/or *questionnaires*. When self-assessing support for the application is needed, this support could include various means such as *textual descriptions or handbooks*, a *software assessment tool, etc*. Since not all of the observed models *provide specific improvement guidelines* when trying to advance from one level to the next, this attribute is included to emphasize the importance. As for the practicality of evidence, implicit improvement activities for future development of the model as well as explicit recommendations are covered. Lastly, some models might need specific training for correct application and if that's the case this training can range from *basic* to *extended*, depending on the level of detail and the desired purpose of use.

4.1.3 Comparison Results

The comparison framework was applied to a variety of maturity models in the PLM area, including the PLM framework proposed by (Batenburg, et al., 2006), the Capability Maturity Model Integration (CMMI), Version 1.3, by the CMMI Institute (CMMI, 2010; Gupta & Rao, 2011), The CMMM model developed in this thesis, the EDEN Maturity Model by the BPM Maturity Model eden e.V. (translated from German) (Allweyer & Knuppertz, 2009) , the enhanced version of Batenburg's PLM maturity model by (Kärkkäinen, et al., 2012), the Knowledge Management Capability Assessment Model by (Kulkarni & Freeze, 2004), the PLM Maturity Reference Model by PLM Interest Group (PLM Interest Group, 2014), the Product lifecycle management model by (Saaksvuori, & Immonen, 2008), the PLM Maturity Model by (Savino, et al., 2012), and the PLM Components Maturity Assessment by (Zhang, et al., 2013).

The comparison results of the *general attributes dimension* are listed in Table 1 - General Attributes Dimension Comparison.

Although the addressed topic might slightly vary, almost all maturity models included in the study can be applied to the PLM domain. Most of the models have an academical root and some have a rather collaborative practitioner-based background. Seven of the ten described models are entirely free to access whereas for the other three there are fees to be paid for a granted access to assessment tools and extensive documentation or professional assistance in the application of the model. Almost all of the models address the management-oriented audience, which verifies once more how important these models can be for the ongoing improvement in organizations.

Although most of the models have different termed business dimensions, similarities and correlations can be found among them. For example, Kärkkäinen's model proposes adding another dimension to the Batenburg model. Concerning the number of business dimensions, there is a vast variety of approaches to be found. Some models focus on only four to five dimensions, coping with the covered subject in a more general matter, while other models show up to 15 or 16 dimensions. The observed maturity levels range from four to six and, similar to the business dimensions, have different labels. For more than half of the models testing is available and has been concluded in different industries. Concerning the CMMI, due to its popularity and widespread acceptance, there are two annual reports that cover the main status of its usage and the maturity of target clients.

General attributes dimension										
Attribute category	**Model**									
	Batenburg (1)	**CMMI (2)**	**CM3 (3)**	**EDEN (4)**	**Kärkkäinen (5)**	**Kulkarni (6)**	**PLMIG (7)**	**Saaksvuori (8)**	**Savino (9)**	**Zhang (10)**
Name	PLM framework for the assessment and guidance of PLM implementations	Capability Maturity Model Integration, Version 1.3	Configuration Management Maturity Model	EDEN Maturity Model for BPM	Defining the Customer Dimension of PLM Maturity	Knowledge Management Capability Assessment Model	PLM Maturity Reference Model	Product lifecycle management	PLM Maturity Model	PLM Components Maturity Assessment
Acronym	-	CMMI-DEV, V1.3	CM^3	EDEN	-	KCMA	-	-	-	PCMA
Primary source	[1]	[10]	[7]	[1]	[4]	[6]	[8]	[9]	[12]	[17]
Addressed topic	PLM	Process Improvement	Configuration Management	Process Management	PLM adoption	Knowledge Management	PLM	PDM/PLM	PLM implementation	PLM Components
Origin	Academic	Academic	Academic	(Academic) / Practitioner-based	Academic	Academic / Practitioner-based	Practitioner-based	Academic	Academic	Academic
Year of	2006	2010	2013	2009	2012	2004	2007	2004	2012	2013
Access	free	free (materials, books available) /charged (e.g. consultants)	free	charged	free	free	charged (access to materials, consultants)	free	free	free
Audience	both	both	management-oriented	management-oriented	both	both	management-oriented	both	technology-oriented	both
Business dimensions (BD)	1. Strategy & Policy 2. Management & Control 3. Organization & Processes 4. People & Culture 5. Information Technology	16 Key Process Areas including: 1. Configuration Management 2. Integrated Project Management 3. Measurement and Analysis 4. Project Planning 5. Risk Management 6.-16. […]	1. Strategy & Performance 2. Process 3. Information Technology 4. Organization & Value-stream 5. Knowledge & Support	170 single criteria within: 1. Goals 2. Strategy 3. Methods 4. Organization 5. Measurement 6. Competencies 7. Communication 8. Documentation 9. IT	1. Strategy & Policy 2. Management & Control 3. Organization & Processes 4. People & Culture 5. Information technology 6. Customer Orientation	Knowledge Capability Areas (KCA): 1. Expertise 2. Lessons learned 3. Knowledge documents 4. Data	1. Data 2. People 3. Processes 4. Technology 5. Knowledge	1. Process 2. Structures 3. IT systems 4. PLM strategy 5. People in PLM change management	15 PLM Components within the 4 TIFO Areas: 1. Technoware-IT 2. Infoware 3. Functionware 4. Orgaware	15 PLM Components within the 5 TIFOS Areas: 1. Technoware-IT 2. Infoware 3. Functionware 4. Orgaware 5. Sustainware
Number of (BD)	5	16	5	9	6	4	5	5	15	15
Maturity levels (ML)	I. Ad Hoc II. Departmental III. Organizational IV. Inter-organizational	I. Initial II. Managed III. Defined IV. Quantitatively Managed V. Optimizing	I. Initial II. Managed III. Standard IV. Optimizing	0. Chaotic I. Rudimentally II. Advanced III. Consistent IV. Controlled V. Sustainable	I. Chaotic II. Conscientious III. Managed IV. Advanced V. Integration	0. Difficult / Not possible I. Possible II. Encouraged III. Enable / Practiced IV. Managed V. Continuously Improved	I. Ad-hoc II. Managed III. Defined IV. Quantitatively managed V. Optimized	I. Unstructured II. Repeatable but intuitive III. Defined IV. Managed and measurable V. Optimal	I. Lowest II. Low III. Medium IV. High V. Top	I. Ad-hoc II. Managed III. Defined IV. Quantitatively managed V. Optimized
Number of ML	4	5	4	6	5	6	5	5	5	5
Testing of the MM	Yes	Yes*	Yes	Yes	Not available	Yes	No access	Not available	Yes	Yes
Number of questions	40	-*	53	No access	-	145	-	-	Not available	Not available
Number of respondents	23	-*	67	No access	-	~750	-	-	~250	Not available
Basic features of the participants' organizations	A: Medium size (15-1000 employees) B: Large size (Over 1000 employees) - Equipment and transport companies (A:3/B:5) - ICT solution providers (A:1/B:5) - Product software companies (A:6/B:1) - Financial service (A:1/B:1)	*As one of the very few maturity models the CMMI has been tested on many occasions in different industries and organizations. The latest testing results found in the 'Maturity Profile Reports' of September 2013 stated around 6,000 concluded appraisals.	Distribution: - 72% in private sector - 28% in public sector Industry sectors: - aerospace (33%) - transportation (12%) - military, defence, government (12%) - automotive (11%) - IT (11%) - others (21%)	-	-	One leading manufacturing company in semiconductor industry - 2 Business Units - All knowledge workers Strong focus on model validation within the organization: - Translation validity - Criterion-Related Validity	-	-	SME's of electromechanical sectors based in southern Italy	Italian prefabrication company (one of the most important companies in the precast and construction field in southern center of Italy)

Table 1 - General Attributes Dimension Comparison

The comparison results of the *design attributes dimension* are listed in Table 2 - Design Attributes Dimension Comparison (Partially confirmed information in brackets). All of the evaluated models present at least the descriptive purpose meaning that they can at least be used to assess the maturity of the organization or certain processes. Seven of them still provide certain prescriptive actions for improvement and only three provide options for benchmarking. Almost all feature process, and more or less object and people maturity concepts, however, only few models can be considered to have covered the customer dimension which is a rather new concept. The composition spreads out diversely throughout all attribute options, but the Likert-like questionnaires/hybrids are utilized more than others. The assessment approach shows a slight focus on staged models, meaning that the maturity of an organization is better observable than specific processes with this selection of models. Only two models are considered multi-dimensional, being able to be applied to almost any domain. Most models are in some way mutable, whether it is by adapting the form, function, or both. Many of the models are verified by some means of testing and only two models are utilized widespread overtime which could be considered completely validated.

Design attributes dimension											
Attribute category	**Attribute**	**Model**									
		(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**	**(8)**	**(9)**	**(10)**
Main purpose	*Descriptive*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Prescriptive	✓	✓		✓	✓	✓	✓		✓	
	Comparative		✓		✓					✓	
Concept of maturity	*Process*	✓	✓	✓	✓	✓		✓	✓	✓	✓
	Object	✓	(✓)	✓		✓	✓	(✓)	✓	✓	✓
	People	✓	(✓)	✓		✓	✓	(✓)	✓	✓	✓
	Customer		(✓)	✓		✓		(✓)			
Composition	*Maturity grid*								✓		
	Likert-like quest./hybrid	✓		✓	✓	✓					
	Others		✓				✓	✓		✓	✓
Assessment approach	*Staged*		✓	✓	✓		✓	✓	✓	✓	✓
	Continuous	✓	✓		✓	✓					
Scope	*One-dimensional*	✓		✓		✓	✓	✓	✓	✓	✓
	Multi-dimensional		✓		✓						
Mutability	*Form*	✓	✓	✓	✓	✓		(✓)	✓	(✓)	
	Functioning	✓	✓	✓	✓	✓	(✓)	(✓)			
Reliability	*Verified*	✓	✓	✓	✓		✓	(✓)		✓	✓
	Validated		✓		✓		✓				

Table 2 - Design Attributes Dimension Comparison

The results of the *usage attributes dimension* are listed in Table 3 below (Partially confirmed information in brackets). Most of the models are created with self-assessment as the appraisal method. Certified practitioners are usually only available with models that are used by institutions or consultant agencies and that come with a charge. The instrument for application divides up quite evenly between all possible attributes, although the for-profit models tend to strive towards audit methodology and using work groups than only document reviews or questionnaires. An often-used supporting tool for the application is a software assessment tool for data analysis and categorization of the results, which has been provided by five of the ten observed models. About half of the models exhibit guidelines for specific improvement activities based on the achieved maturity level. Almost all of the models show implicit improvement activities. The adaption to the dynamically changing requirements is very relevant and important for future usage purposes. One downside to the for-profit models is the required training for the auditors.

Usage attributes dimension											
Attribute category	**Attribute**	**Model**									
		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Method of application	*Self-assessment*	✓	✓		✓	✓	✓	✓	✓	✓	✓
	Third-party assisted assessment		✓	✓							
	Assessment by certified practicioners		✓		✓			✓			
Instruments for application	*Document reviews*		✓		✓	✓			✓		
	Focus groups		✓		✓			✓			
	Questionnaire	✓	✓	✓	✓		✓			✓	✓
Support of application	*No supporting materials*					✓	✓		✓	✓	✓
	Textual description or handbook		✓		✓			✓			
	Software assessment tool	✓	✓	✓	✓			✓			
Guidelines for specific improvement activities	*Not provided*			✓		✓	✓		✓	✓	✓
	Guidelines provided	✓	✓		✓			(✓)			
Practicality of evidence	*Implicit improvement activities*	✓	✓	✓	✓	✓	✓	✓	✓		✓
	Explicit recommendations		✓		✓						
Required training	*None*	✓		✓		✓	✓		✓	✓	✓
	Basic		✓		✓			✓			
	Extended		✓								

Table 3 - Usage Attribute Dimension Comparison

4.1.4 Reflection and Conclusion

The focus of this chapter was to try and provide a maturity model comparison framework. Past categorization and comparison approaches have thoroughly been analyzed to develop a framework that covers all important attributes, which are needed to identify and contrast different maturity models. The aim of the comparison was not to rank the models but rather reduce the search time for specific models, allow easier communication, identify differences and similarities, and last but not least, provide a comparison of structure between well known and widely used models as well as novice and improving models. Besides serving its purpose in this thesis, this framework is designed to be used as a standalone benchmark for future maturity model developments as well.

One reflection to the framework could be that since the comparison only covers a qualitative evaluation, a weight based scoring system could be implemented to add further detail to the analysis and to be able to rate the models more precisely with respect to their intentional application. This was considered when developing the model and since the attributes are not necessarily rankable (e.g. having a certain assessment model is not necessarily superior to others), it was decided to keep the qualitative nature for the users to decide on the results based on their requirements. Another point that could always help models like this is to include as many models as possible and because the PLM area is generally becoming more important to companies and organizations, more models could benefit future research and application further.

Reflecting on the primary purpose of the developed framework, the results of the comparison clearly provide valuable information to reflect on the structure of the proposed Configuration Management Maturity Model and its usability. Overall, based on the outcome of the comparison, this maturity model in many ways could serve its intended purpose in a relatively comparable manner as to the well-established models. This is mainly owed to the development of this model being based on a formal maturity model design structure.

As for improvement areas, even though the model is tested in a global assessment study (explained in the next section), it could for sure benefit from further practical implementations in a detailed audit-based structure. This would require detailed KPIs for each dimension and its respective sub-categories. Such KPIs, if extracted from valid sources, could enhance the performance of the model to cover the greatest details in a quanititative manner for each individual organization assessment which is the ultimate goal of the model.

4.2 Cross-industry Configuration Management Maturity Assessment

In a partnership with a German consulting company[1], one of the highly active global organizations in the area of Configuration Management consulting, the model was utilized and tested to evaluate the maturity of a large number of organizations. For this purpose, an appraisal material was developed with the aim of assessing organizations' CM practices against CMMM. In order for the appraisal material to be distributed rapidly and easily to all target cases and at the same time collect as much detailed information as possible, the online survey methodology was selected. The structure and results of this cross-industry maturity assessment study is explained in the following sections.

4.2.1 Survey Design and Structure

The survey was developed through several iterations in collaboration with subject matter experts from the partner company. For enhancing the quality and user-friendliness, the survey was made available in both English and German languages. Prior to distribution, the survey was naturally pre-tested by experts who were not involved in the initial phases of the review process.

The survey consisted of 53 multiple choice questions, covering different capability areas in the maturity model and the extent of their penetration in organizations' business processes. The body of the question reflected the sub-categories under each dimension while the multiple choices matched with the generic maturity levels. This approach was chosen to give a logical measure for the respondents for rating the maturity of their organization in each respective subject. The average response time was estimated in the pre-test to be about 20 minutes and the distribution of the questions could be observed in the figure below.

Total Questions	General	Demographic	Strategy & Performance	Processes	Information Techonology	Organization & Value-stream	Knowledge & Support
53	6	8	4	13	10	6	6

Figure 21 - Distribution of questions in the designed online survey

The survey was distributed among 150 selected CM-related professionals from different industrial sectors who were close clients of the consulting group. During the one-month data collection period, a total of 67 full responses were

[1] P3 Group Consulting

collected. Questionnaire design can be found in Appendix A. The comprehensive results of this study is analysed and presented in the following sections.

4.2.2 Study Results

4.2.2.1 General Information and Demographics

In this section, general information about the participants and their organizations are provided. More than two third of participants are from private sector where resources are limited and standardized practices could make a great impact on the survival and success of the organizations.

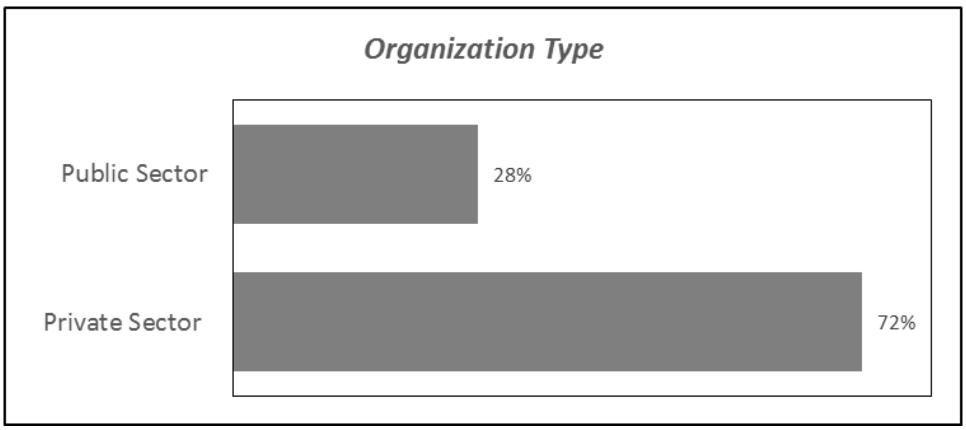

Figure 22 - Distribution of Organizations With Respect to Their Sectors

More than 50 percent of the participants originated from large enterprises where the activities are distributed mostly among various geographical locations and the importance of an intra-organizational standardization and integration is prominent. Approximately 44% of the participants were part of SMEs (10 to 1000) where some level of operational flexibility exists but there is a need for inter-organizational standardization and integration.

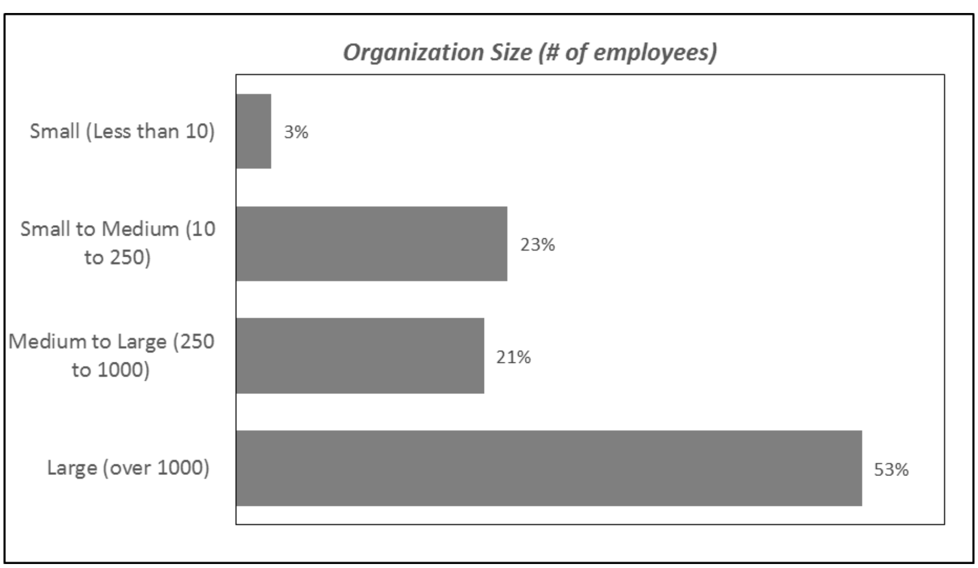

Figure 23 - Distribution of Organizations With Respect To Their Size

The distribution of industry sectors among participating organizations are rather diverse. For the purpose of meaningful information inference, industry-based analysis in this study will be performed on the top five organizations with respect to participants; namely, Transportation, Information Technology, Automotive, Aerospace and Military & Government.

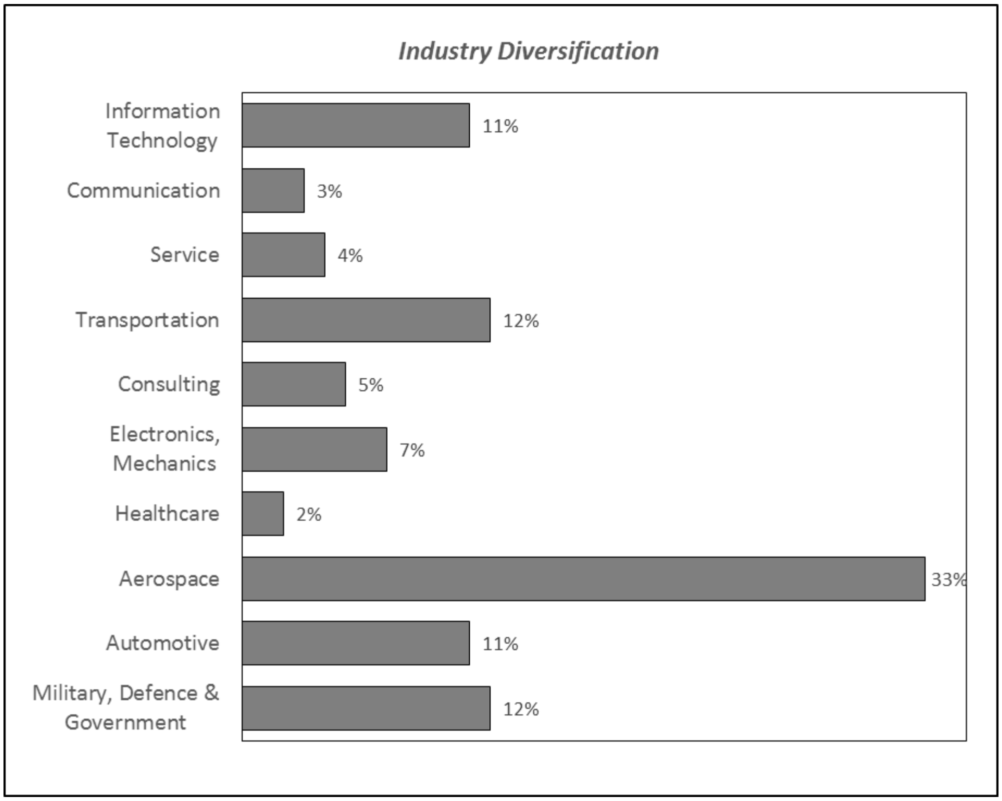

Figure 24 - Distribution of Organizations With Respect to Their Industrial Sectors

To ensure the reliability of collected information and confirm the level of understanding of the subject among participants, information about their extent of involvement with CM practices in their roles was collected.

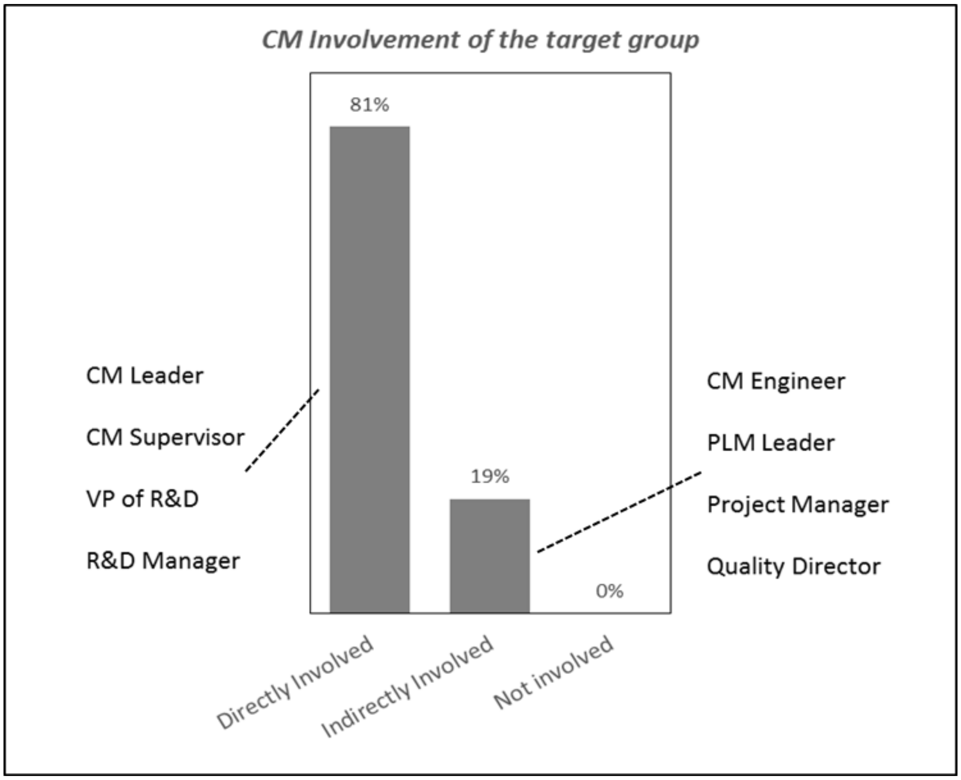

Figure 25 - Position and Level Of Involvement of Participants In CM Activities

When participants were asked about previous maturity assessments of CM in their organizations, it was realized that only 28 percent had used assessment frameworks for this discipline. This leaves the majority of organizations less aware of their CM activities and could to some extent revalidate the focus of this research.

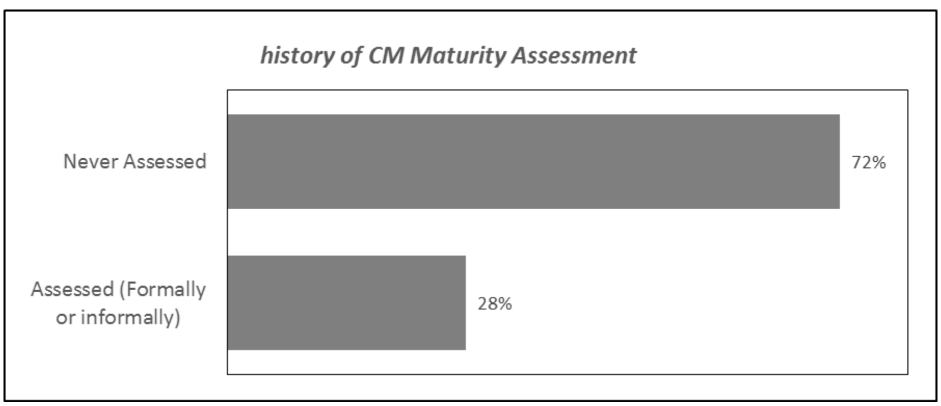

Figure 26 - CM Assessment History

4.2.2.2 Dimension-based maturity

4.2.2.2.1 Strategy & Performance

As described before, the strategy dimension of CM maturity model includes subjects such as creation of CM strategy and policy, deployment of this strategy in different organization units and measuring the CM performance by utilizing KPIs. The following figures illustrate the maturity of various industries in this dimension as well as overall maturity of each sub-dimension.

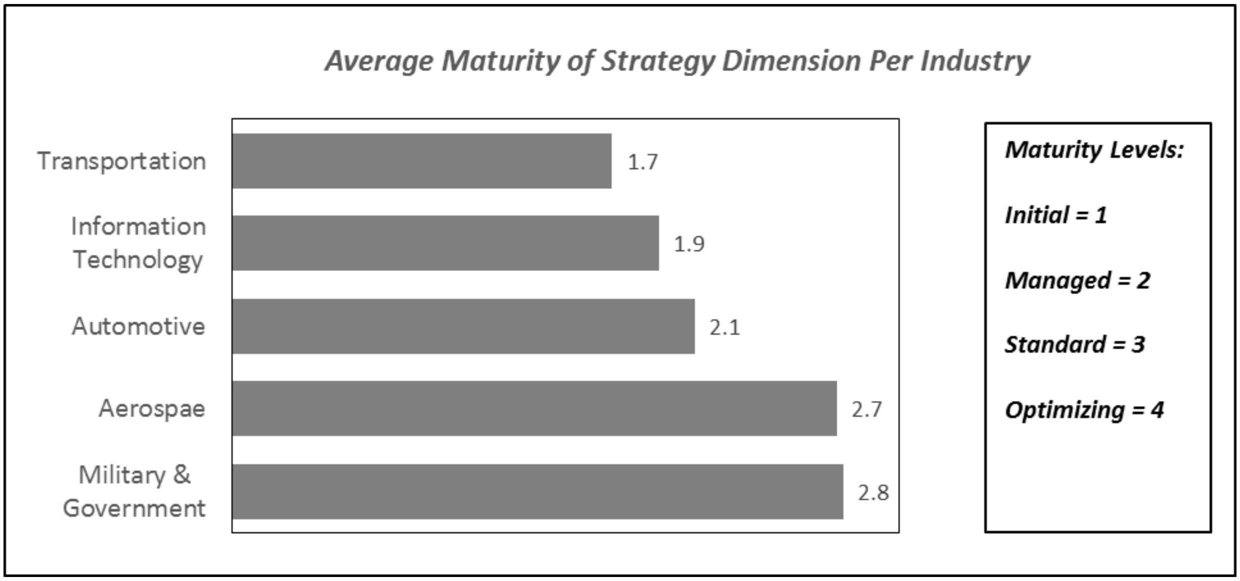

Figure 27 - Average Industry-Based Maturity of Strategy Dimension

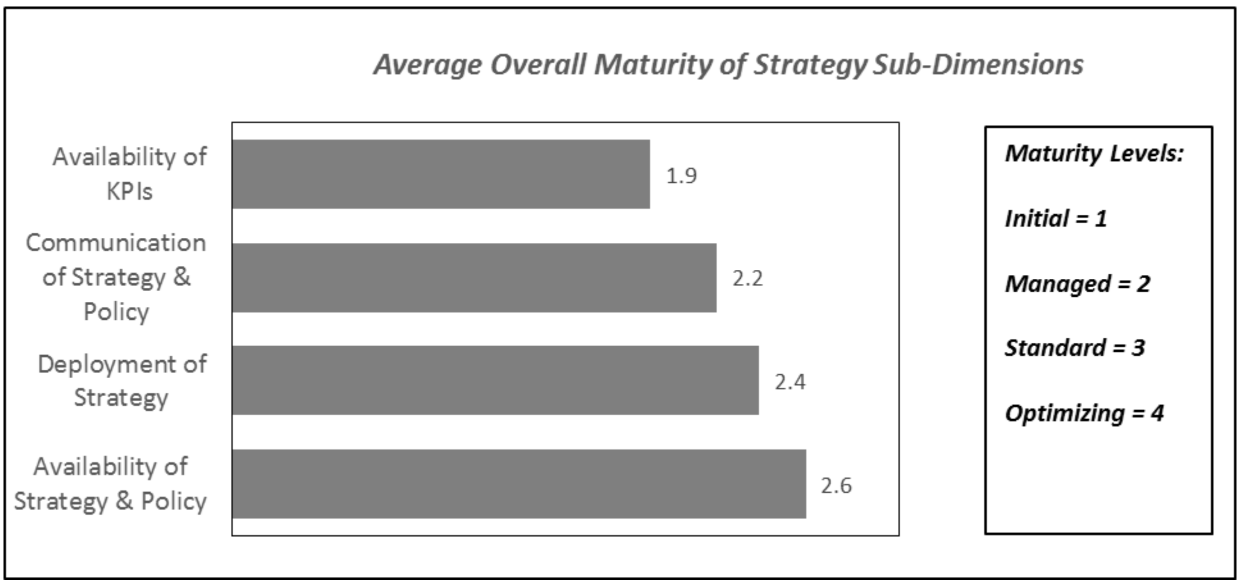

Figure 28 - Strategy Sub-dimensions Maturity among All Participants

The highest potential lies in the development and utilization of suitable KPIs for CM performance measurement as well as communication of the CM strategy to all stakeholders.

4.2.2.2.2 Process

The Process dimension of the CM maturity model includes subjects such as standard CM processes including Configuration Identification, Change Control, Status Accounting and Audit. Additionally, the management and maintenance of processes, accessibility of processes, process ownership and customizability of CM processes are part of this dimension. The following figures illustrate the maturity of various industries in this dimension as well as overall maturity of each sub-dimension.

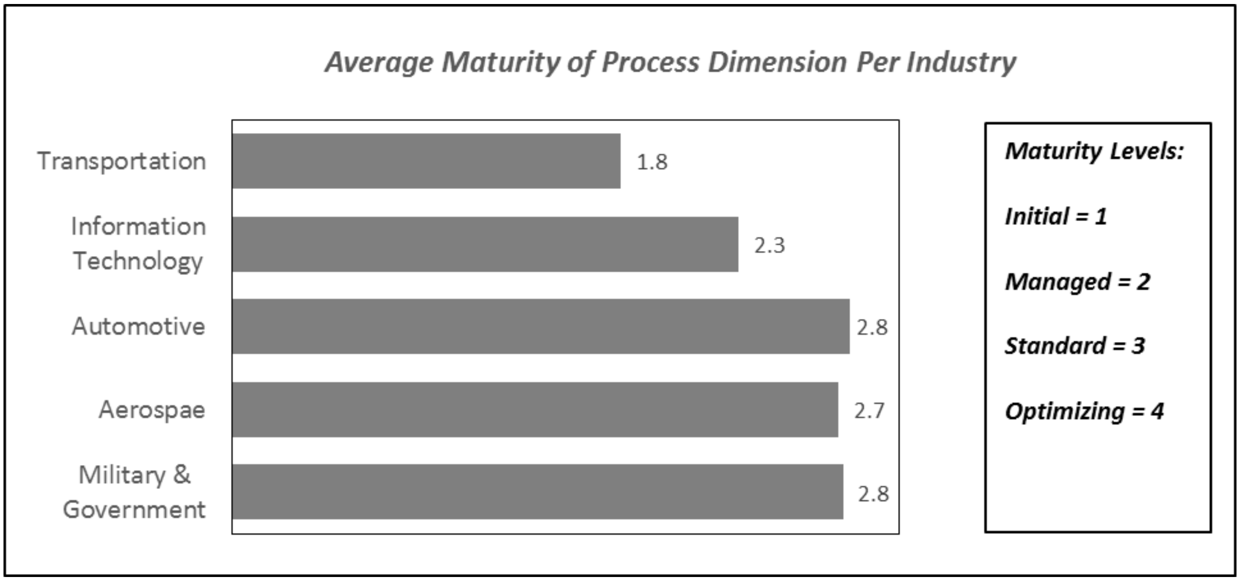

Figure 29 - Average industry-based Maturity of Processes Dimension

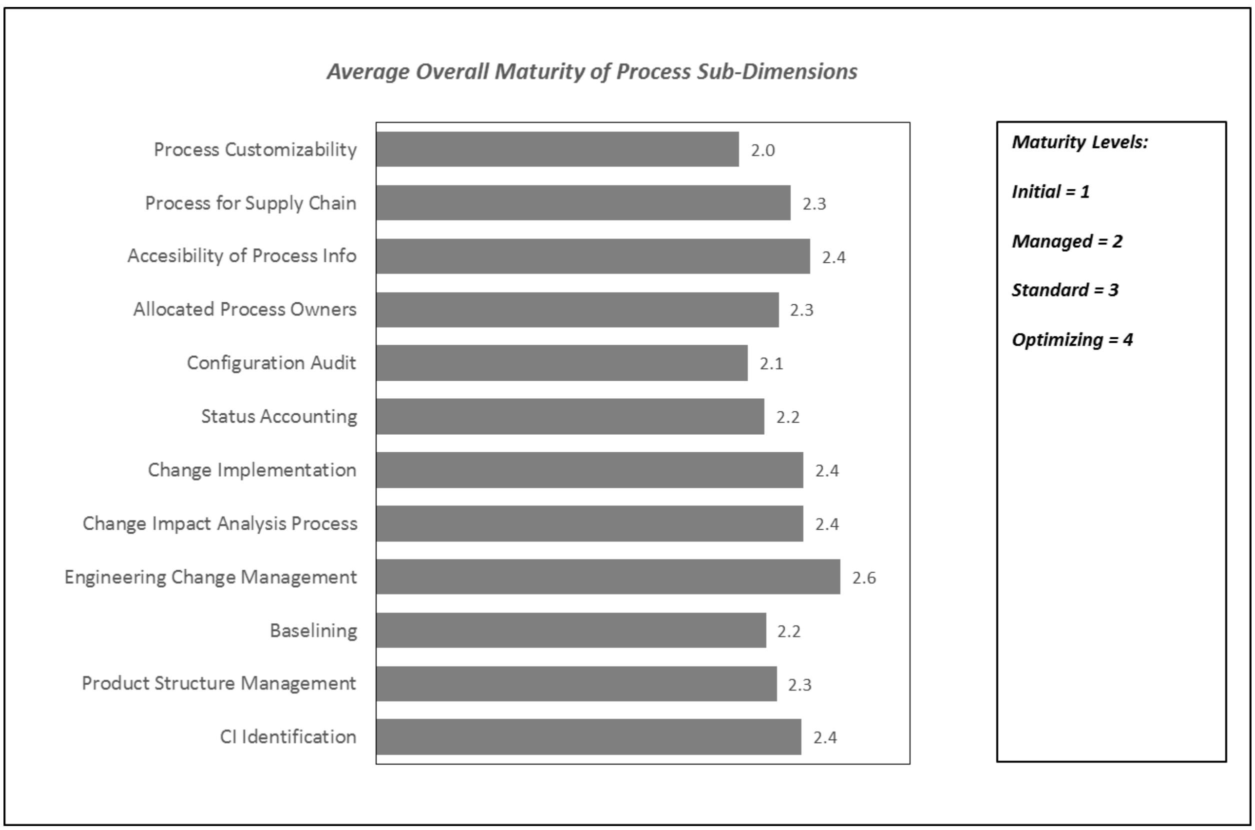

Figure 30 - Process Sub-dimensions Maturity among All Participants

The highest potential for improvement among the various processes can be identified in the process interface between CM and supply chain, as well as the Configuration Audit process. Industry experience shows that the integration of suppliers is a big challenge to manufacturers and integrators of complex products. Furthermore, the measuring of a company's CM process quality is often neglected and therefore improvement potential remains hidden.

4.2.2.2.3 Information Technology

The IT dimension of the CM maturity model includes subjects such as the integration of CM IT systems with main IT systems, user-friendliness of CM IT system, support of IT systems in change management and accessing and managing work-flows, as well as the authorization capabilities of the CM IT system. The following figures illustrate the maturity of various industries in this dimension as well as overall maturity of each sub-dimension. The results suggest that the Aerospace industry has the highest average maturity of about 2.7 which is close to the "Standard" level. As can be observed, the Transportation industry has a maturity of 1.13 which represents high potential for improvement for this industry in the IT dimension.

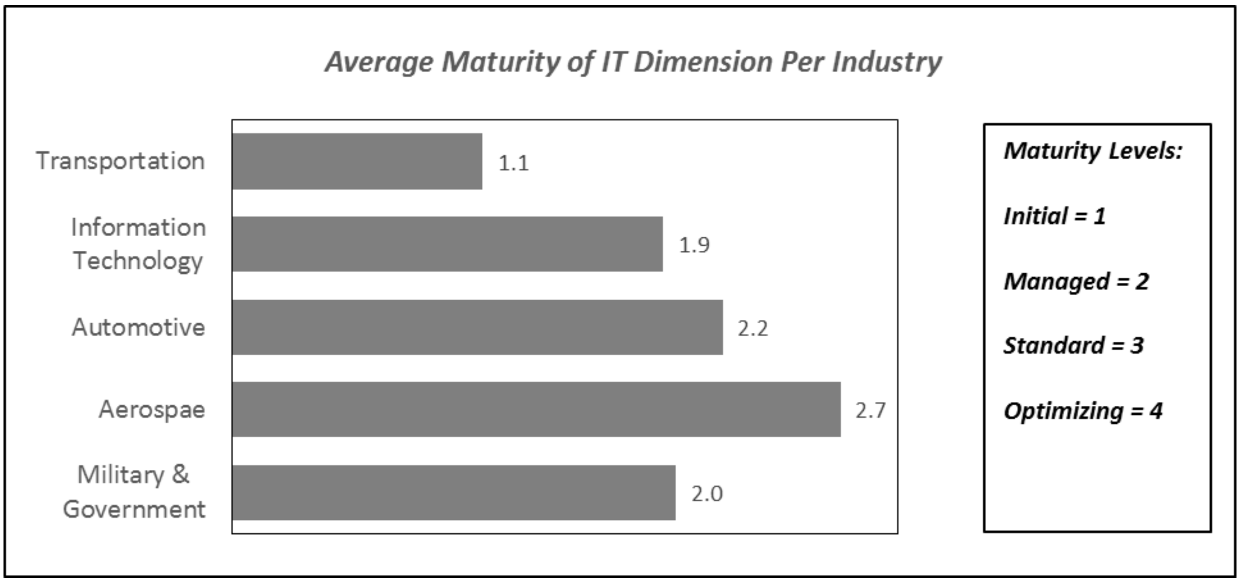

Figure 31 - Average Industry-Based Maturity of Information Technology Dimension

Figure 32 - Information Technology Sub-Dimensions Maturity Among All Participants

The results suggest a need for improvement with respect to the IT support for Engineering Change Management processes.

4.2.2.2.4 Organization & Value-stream

The Organization & Value-stream dimension of the CM maturity model includes subjects such as CM organization structure, roles and responsibilities, cross-functional collaboration, sub-contractors activities as well as the involvement of customers and stakeholders in CM. The following figures illustrate the maturity of various industries in this dimension as well as overall maturity of each sub-dimension.

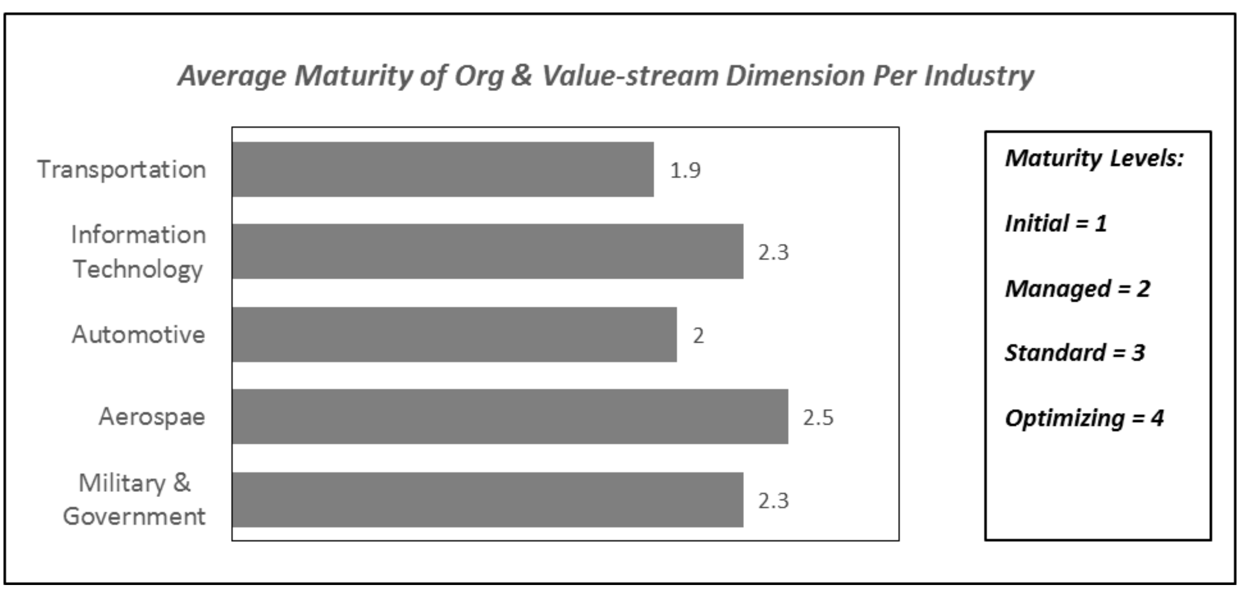

Figure 33 - Average Industry-Based Maturity of Organization & Value-Stream Dimension

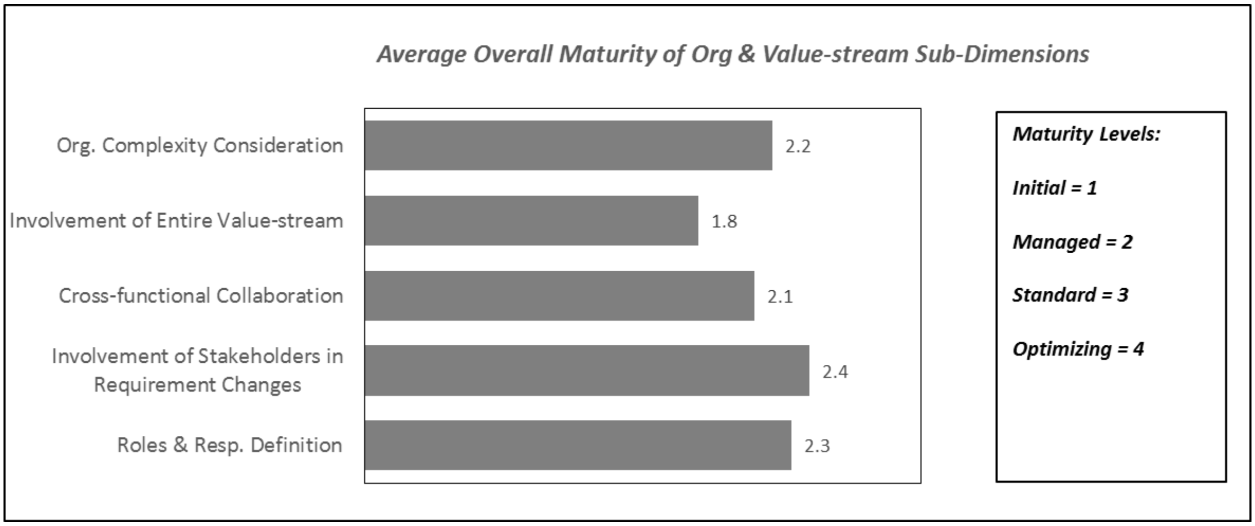

Figure 34 - Organization & Value-Stream Sub-Dimensions Maturity Among All Participants

The involvement of entire value-stream such as suppliers and sub-contractors is the highest improvement potential in this dimension. This clearly reflects the challenges which are coupled with the risk-sharing and outsourcing approach by utilizing a global supply chain across product lifecycle.

4.2.2.2.5 Dimension 5: Knowledge & Support

The Knowledge & Support dimension of the CM maturity model includes subjects such as the utilization of standard CM terminology, CM trainings, accessibility of knowledge sources and best practices, CM promotion by top management, communication of CM benefits to stakeholders as well as external and internal benchmarking. The following figures illustrate the maturity of various industries in this dimension as well as overall maturity of each sub-dimension.

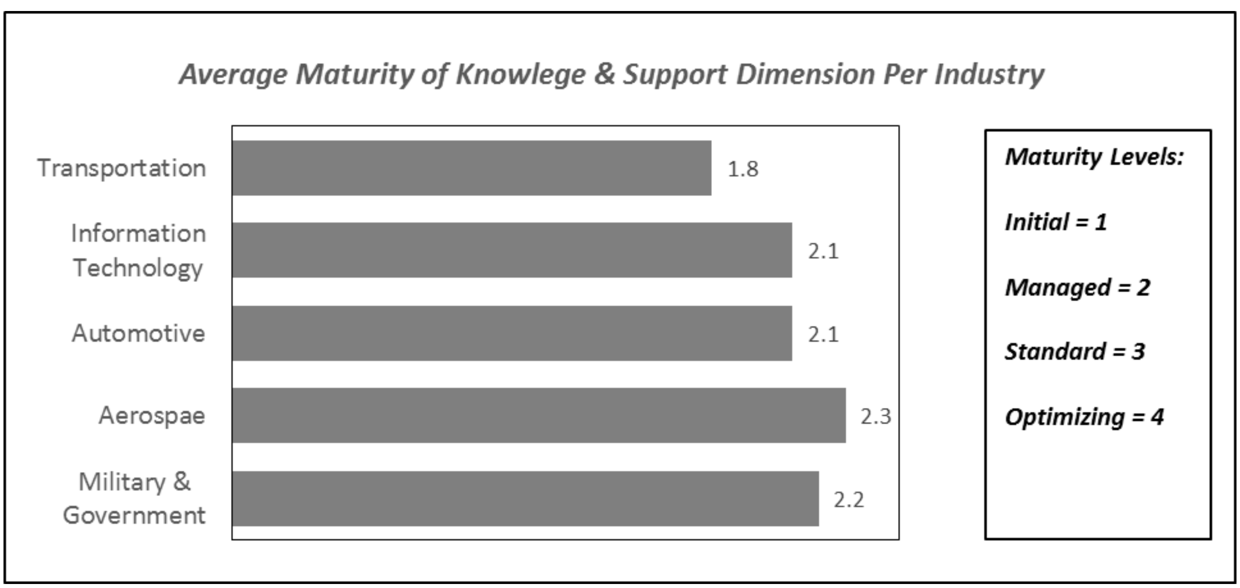

Figure 35 - Average Industry-Based Maturity of Knowledge & Support Dimension

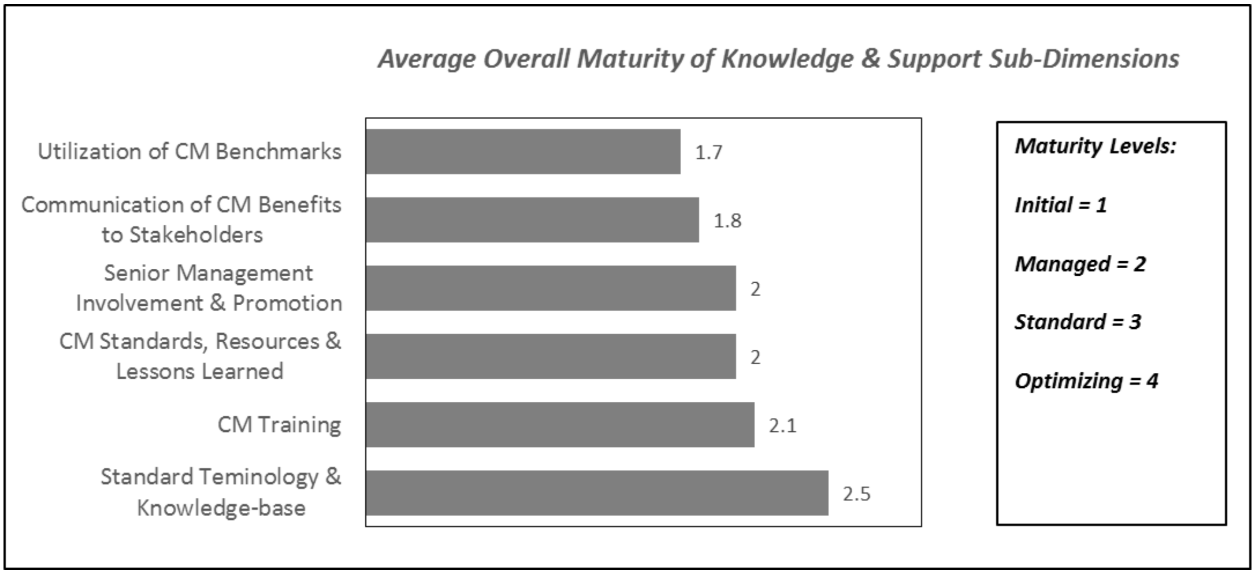

Figure 36 - Knowledge & Support Sub-Dimensions Maturity Among All Participants

CM benefits are hardly known in all departments of every company. In fact, industry experience shows that, despite the fact that CM is a holistic discipline there is a stronger concentration of CM knowledge in the engineering departments. Improvement potential can only be fully exploited if CM is promoted across the whole value-stream and cross-functional contributions are received.

4.2.2.3 Industry-based comparison

The rather substantial difference among various industries' CM maturity could be explained by the sensitivity level of products from a CM standpoint, e.g. high sensitivity in the Aerospace and Military industries. However, there is still room for improvement in all industry sectors to reach "Standard" and "Optimizing" levels.

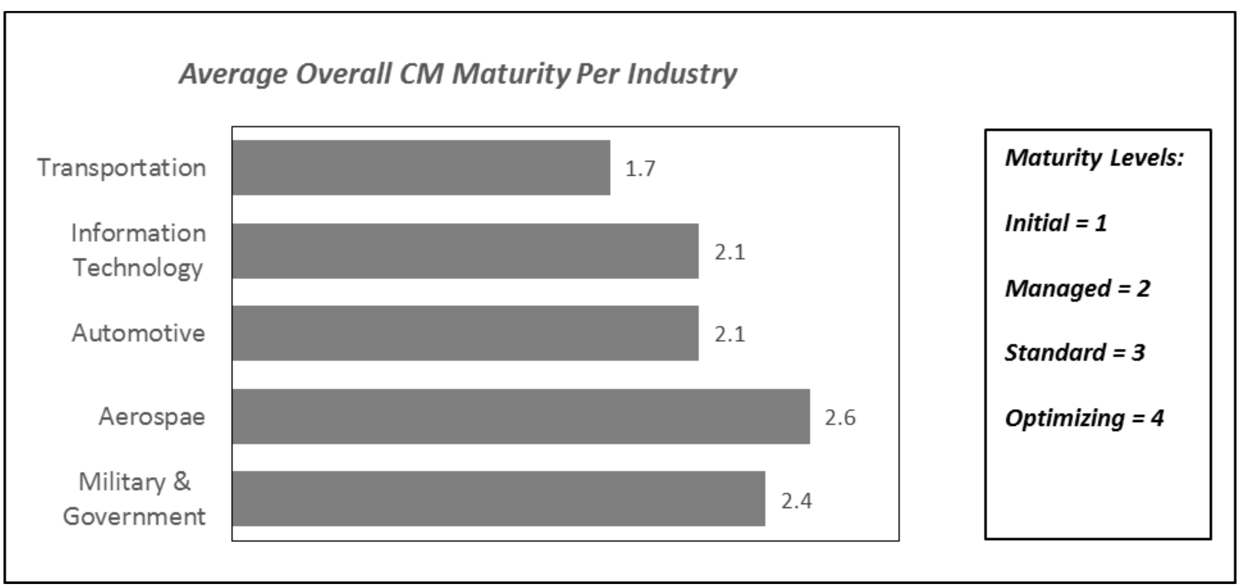

Figure 37 - Cross-industry Maturity Comparison

4.2.2.4 Size-based comparison

When the maturity is compared with respect to organization size, a rather considerable overall difference in all dimensions can be observed.

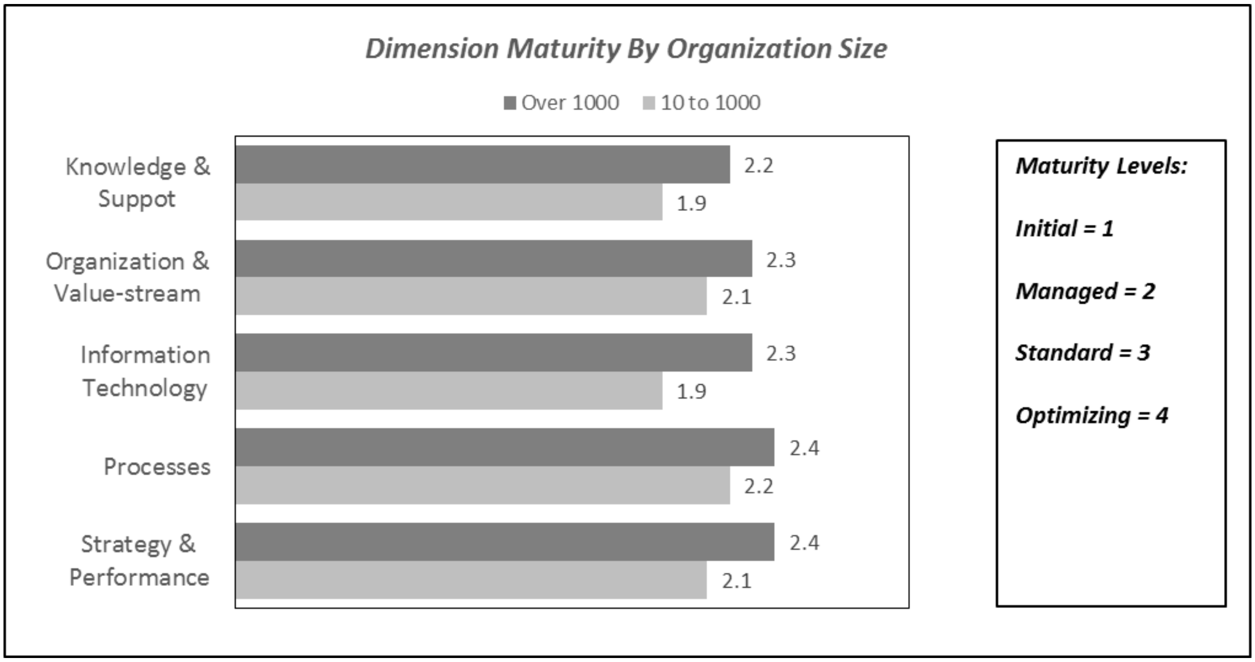

Figure 38 – Overall Maturity of Large Organizations

This observation was confirmed from industry experience, with the following possible explanations:

- SMEs with growing responsibility and product complexity, such as system suppliers to the Aerospace or Transportation industry, are facing substantial challenges to enhance their CM processes, information systems or knowledge transfer. Often, the capacity to do so is very limited and a clear priority is given to the operational business.
- Regarding the information technology support for CM, large companies frequently implement extensive IT solutions with a broad range of CM functionality, such as PLM and ERP systems. In SMEs, the use of IT systems often has a pragmatic background with self-programmed or open-source solutions. Here in many cases, CM functionality is iteratively added to the existing systems.
- Configuration Management roles in SME organizations are primarily allocated to resources, which at the same time fulfill other roles, such as project managers, quality managers or design engineers. Large companies often have the capacity to define organizational units with dedicated resources to fulfill CM tasks which is proven to generate better results.

4.2.3 Potential Improvement Areas

The findings of this study could point out the areas where the overall maturity of all industries are rather low and there is a need for further academic support to extend the variety of solutions and best practices. The first step after maturity assessment is to identify under-developed areas relative to areas with average or advanced maturity. Since the highest maturity of a discipline is as high as the least mature module within that discipline, in this section the sub-dimensions that have overall maturity lower than managed or level 2 are further discussed for possibility of research contributions in the field.

4.2.3.1 Senior Management Support & Performance Indicators

The senior management support and promotion is an influential factor in the success of any discipline and CM is not an exception. However, one of the challenges in the case of Configuration Management is the support nature of this discipline where the success of projects are rarely associated with its successful implementation. A possible reason could be the lack of generic performance measurement activities in this field which based on the results of this study is one of the main areas for improvement. Therefore identifying key performance indicators to measure CM performance could very well increase the chance of identifying visible correlations with project performance and could provide the basis for senior management to acknowledge the importance of this discipline.

4.2.3.2 Value-Chain Involvement

The involvement of the entire value-chain in standard CM processes and tools tends to be one of the main areas for improvement. Outsourcing responsibilities, such as design and development or maintenance to a global supply chain lowers cost and risk to a certain extent. Nevertheless, internal complexity increases in terms of organization, processes and tools along the Product Lifecycle. Consequently, Configuration Management needs to include suppliers appropriately in terms of CM roles, CM process interfaces as well as tool interfaces. Vice versa, suppliers must strive to create their CM environment to be as flexible as possible in order to adapt to customer processes and tools. The overall trend in the IT industry to provide global and standard tools to support all lifecycle phases and value-chain players, could decrease this challenge in the near future. For sure, this shall be supplemented by internal organizational enhancements and centralized CM roles to assure the information consistency and process standardization.

4.2.3.3 Information Technology Support for Engineering Change Management

For an efficient application of CM throughout Product Lifecycle, it is essential to have appropriate Information Technology support. The results show a particular potential for improvement regarding IT support specifically in the area of Engineering Change Management. As technical changes can occur at any point in Product Lifecycle, a well managed information system that could sufficiently support the complex decision making processes, could be highly beneficial and assist organizations in their efficiency and effectiveness.

4.2.3.4 Utilization of CM Benchmarks

As it is the case with many disciplines, utilization of internal and external benchmarks expand the improvement opportunities and consequently competency of the organization in that discipline. As it was reconfirmed in the general information section, more than 70% of participating organizations had never assessed their CM practices. This implies that there is not a clear understanding of strength and weakness on those organizations and therefore, there is a very low chance for those organizations to benchmark best-in-class organizatiosn or even internal departments. In addition to industry wide assessment like this study that could point out several improvement areas, as discussed in the previous sections, utilization of internal KPIs could provide the means of internal comparisons among departments or projects to find best practices and exploit lessons learned.

4.2.4 Maturity Model Reflection

As it was discussed in the introduction of this cross-industry assessment, one of the main intentions behind this study in addition to identifying potential focus areas for research and industry, was to evaluate the level of usability of the developed maturity model. For this purpose, participants of the study were asked to rate the level of coverage within this model. The results of this rating could be seen below.

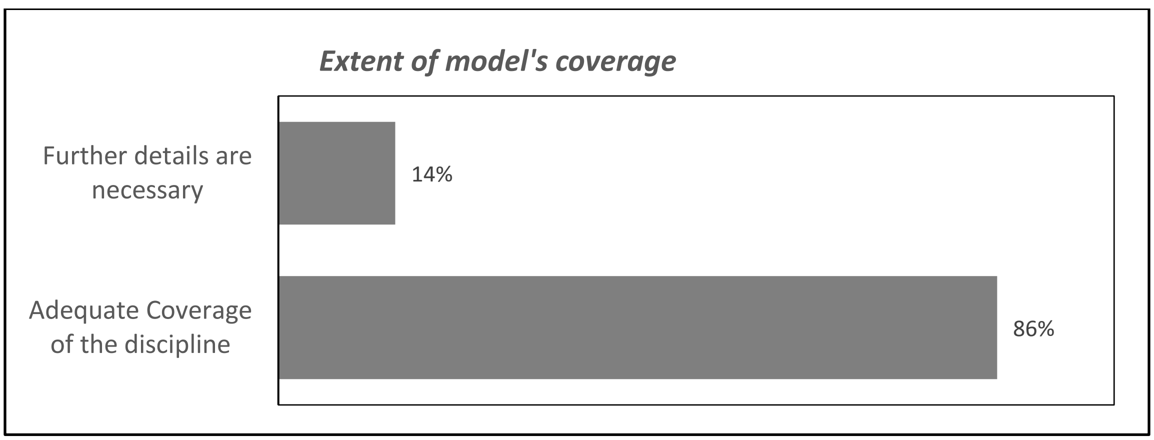

Figure 39 - The Participants' Opinion on Level of Model's Coverage on CM Aspects

Additionally, the results of the survey was reviewed with four subject matter experts in different industries where all acknowledged the main findings to be in compliance with what they experience as improvement opportunities in their organizations.

4.3 Optimization Opportunities

As the results of the analyses in previous chapters suggest, there are several opportunities to both enhance the functionality of the maturity model and contribute to the configuration management research as a whole. Due to the time and resource constraints facing this research, the author focused on two primary areas for further research.

First, in order to enhance the functionality of the developed maturity model further, research was performed with a focus on CM key performance indicators. According to the results of the comparison framework in section 4.1.4, detailed KPIs could better equip the model with quantitative means of maturity assessment and lay the foundation for enhancing the model functionality to more prescriptive purposes. Meanwhile, such KPIs could very well be adapted by CM practitioners in their efforts towards their CM strategic deployment which itself was identified as one of the less mature areas in the industry.

Second, the support solutions for Engineering Change Management, process was identified as one of the areas requiring further research attention. After a thorough gap analysis by literature review and empirical studies, a new decision support concept was designed and proposed that could for sure assist the CM practitioners in their ECM decision making process. As the results of the cross-industry maturity assessment as well as empirical interviews show, successful implementation of such concepts that support the ECM process, could very well enhance the maturity of CM in all industries.

In sections 4.3.1 and 4.3.2, brief explanations regarding the two concepts, their development methodology and final results are provided. Comprehensive research in each of the areas are illustrated in the following chapters.

4.3.1 Key Performance Indicators

With the intention of developing suitable CM performance indicators, difference dimensions of CMMM were used as critical success factors. Therefore, only the quality of CM-activities concerning the given sub-dimensions is the point of interest. Therefore, measures which focus on the overall business performance of a company or an organizational unit (e.g. Return on Investment) were excluded. Quality in the context of the indicators refers to the question: “How ‘good’ is the activity X performed?”

In order to get a first list of possible Performance Indicators for the different CSFs, a broad literature analysis was done. Based on this first collection, semi-structured interviews with industry experts were performed to both validate the proposed indicators and adjust them with respect to their feasibility of application, data availability and importance. The interviews were held with four experts from automotive, aerospace industry, plant engineering industry as well as engineering consulting. As a result of the interviews some of the indicators have been approved without any further remarks, some have been supplemented for a better understanding and match with currently available indicators, and a few have been excluded because of doubts about the general significance.

4.3.2 ECM Decision Support System

Based on the results of the maturity assessment it was identified that the area of ECM information modelling and IT support could very well be a potential for further research.

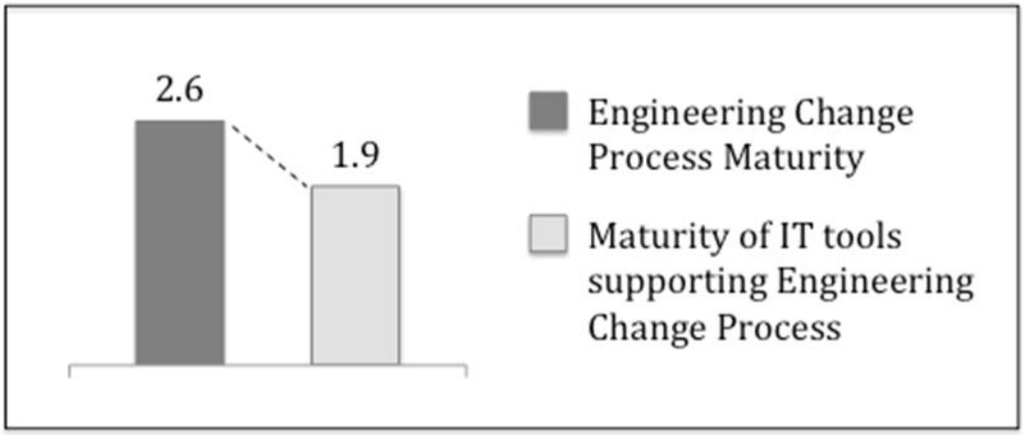

Figure 40 – ECM Process Vs. IT Support Maturity

Further gap analysis in the field clarified one of the main point of focuses to be the change approval decision making process. It was confirmed by several practitioners especially in the complex engineering environments such as CERN, that the decision making process suffers from lack of opportuinity to consider more variables than is possible for human mind. Since, the need for a support system to provide feedback by better data analysis became prominent, further research was performed to analyse the available information models in the field. After a thorough model comparisons and extensive literature research in several areas of Engineering Change Management, Decision Support Systems and Information Modeling, a new conceptual solution was designed and proposed to utilize the change information that is gathered from the first day of product or facility design for better decision making on any engineering modification during the long lifecycle of that product or facility.

5. Configuration Management Key Performance Indicators

5.1 KPIs in the Strategy Implementation Realm

KPIs represent a set of measures focusing on those aspects of organizational performance that are the most critical for the current and future success of the organization (Parmenter, 2007). In any department or discipline within an organization, there is a need to measure success and thrive for excellence. KPIs are the tools to simplify this measurements and if designed thoughtfully, they could act as light in the dark and be driving forces for companies in their success. KPIs in general are used for monitoring and improving performance of an organization with respect to a focus area. (Klingebiel, 2013) highlights the importance of having the right set of KPIs for performance measurements by pointing out three main perceptions:

a) What can't be measured can't be managed

b) What gets measured gets done

c) Measurement influences behavior

In order to develop successful KPIs for the purpose of this research, there is a need for understanding how KPIs contribute to the success of an organization strategy. In the strategy alignment pyramid provided by (Bauer, 2004), regardless of the focus area, there is a clear understanding and logic behind every deployment level of strategy within an organization.

Figure 41 - Strategic Hierarchy

When the same logic applies to the CM discipline, the vision and strategy could clearly match the maturity enhancement and improvement activities towards reaching success in this field. With the objective of profound and complete application of various aspects of this discipline, the critical success factors (CSF) match the primary aspects of this discipline. Since such primary aspects are already captured extensively as dimensions and sub-dimensions of the CM Maturity Model, those are used as the basis for the KPI development. As such measureable KPIs could privde organizations with clear gaps and focus areas, Key Action Initiatives could later on easily be generated from each KPI and their benchmarked acceptable range. This will be the basis for a prescriptive maturity model where it could simply provide action plan and guidelines based on identified improvement potentials.

5.2 CM Maturity KPIs

In order to get a first list of possible KPIs for the different sub-dimensions, a broad literature analysis was performed. The outcome then was reviewed through semi-structured interviews with industry experts to both validate the indicators and adjust them with respect to their feasibility of application, data availability and importance. The interviews have been done with four senior experts, from automotive, aerospace, engineering and consulting industries to cover a broader spectrum. As a result of the interviews, some of the indicators have been approved without any further remarks, some have been supplemented for a better understanding, and a few have been excluded because of doubts about the general significance and possibility for information retrieval. At the same time, a few KPIs that were practiced by industry were also proposed by subject matter experts and were included in the model. The final results were categorized and structured in the main format of CM Maturity Model dimensions and sub-dimensions and could be found in the following table.

Performance Indicators:	Initial source:	Expert Comments:
% - percentage	[#] : Reference	V : Validated
# - number/amount of	[I] : Interview	U : Updated
Ø - average		E : Emerged

5.2.1 Strategy& Performance

CM Strategic Objective and Policy: *For successful CM application, its objectives and policy should be developed in correspondance to the corporate strategy*		
- % to which CM strategic objectives are relevant to the overall corporate strategy: *the CM strategy needs to fit to the company's overall goals to support their achievement*	(Hass, 2003) (Robbins, et al., 2013)	V
- Actual frequency/target frequency of updating CM objectives: *CM objectives need to be rechecked regularly to be adapted to new corporate goals*	(Hass, 2003) (Rohm, 2008)	

- % to which CM policy & objectives are covering all main CM activities in the organization: *proper policy and objectives should cover all important subjects to avoid confusion and constitute a guideline*	(Ulrich, 2001) (MIL-HDBK-61 , 1997)	U V
Deployment of CM Strategy in Different Organization Levels: *CM strategy needs to be deployed to all levels to provide clear objectives for each team*		
- % of deployed goals, which can be clearly linked to a higher CM objective - # CM units having a mission statement that clearly defines their objectives and mission: *it is important that all organizational units know what they need to do to achieve main CM objectives*	(Robbins, et al., 2013; Martin, 1990) [I]	V E
Communication of the Deployed Strategy to Stakeholders: *To be able to support CM objectives, the strategy has to be effectively communicated to the stakeholders*		
- # of strategic deviations associated to communication: *communication quality can be assessed by conducting audits.* - % employees who can explain the CM strategy and what it means in terms of their daily work: *if employees know how their work contributes to CM success the communication has worked properly* - Effectiveness of policy implementation: the policy should not only be mentioned on a document but it should be understood and applied in the operative business	(Barret, 2002) (Barret, 2002) (Martin, 1990)	U V V
KPIs for Performance Measurement: *The fulfillment of CM objectives need to be measured with Key Performance Indicators*		
- Quality of data available for KPI results (sufficient quantity, completeness): *if the database for the KPI application is not sufficient, then the KPI results won't be reliable* - % KPIs linked to strategic objectives and critical success factors: - Frequency of KPI reviewing and improvement: *the usefulness of a KPI should be checked regularly with respect to applicability and need*	(Watts, 2009) (Eckerson, 2006) (Maier, 2004) (Parmenter, 2007) (Eckerson, 2006)	V V U
Regular Measurement of KPIs: *KPIs needs to be measured regularly based on clean and useful information*		

- Frequency of measurement and reporting: *depending on the scope of the KPI and their respective data collection horizon, the measurement needs to take place constantly and results be reported*	(Eckerson, 2006) (Watts, 2009)	V

5.2.2 Processes

Clear processes for different org. units, projects and lifecycle phases: *Processes should be defined, transparent, controllable, effective, efficient, straightforward, robust and at a right level of detail.*		
- # of requests about processes due to a lack of understanding: *clear processes should be defined in a comprehensible way*	(Fischermanns, 2009)	U
- % cases where the result of the process is constructive in a way that predefined objectives are achieved	(Fischermanns, 2009; Becker, 2008)	U
Standard processes: *Standard CM processes like configuration identification, baselining, product structure management, change evaluation, control & implementation, status accounting and configuration audits.*		
- % of processes with clear SOPs	[I]	E
- Ø time to approve Engineering Changes with respect to complexity	(Watts, 2009; Lyon, 1999; Watts, 2008)	V
- Ø cost per change	(Watts, 2009; Lyon, 1999; Watts, 2008)	V
- # non-confirmities in configuration of the final product: *all standard processes are trying to prevent these deviations*	[I]	E
- Frequency of reports and audits	(Lyon, 1999)	V
- Ratio of successful engineering changes after approval over total	[I]	E

Process ownership, maintenance and update based on feedbacks: *one specific role is responsible for a process - processes are maintained, updated and improved - feedback among the employees is encouraged and rewarded.*		
- Frequency of feedbacks: *more reasonable if combined with the benefit* - % of processes with a specific and active process owner, who is responsible for process audit and the interfaces with adjacent processes - % feedback from each lifecycle stage: *Ideally feedback from initial phases are less costly and enhances the functionality of the process before implementation*	(Fischermanns, 2009; Becker, 2008) (Fischermanns, 2009) [I]	U U E
Stakeholder access to processes: *Access of stakeholders to process information must be defined and managed in an efficient way*.		
- Ø Time taken for stakeholders to find information about standard processes: *In some organizations this is controlled by the number of contact points reached before finding explanation* - % of satisfied stakeholders regarding accessing process information	[I] (Bourne, 2009)	E U
Process customizability for various requirements: *Processes should be able to handle different requirements in a continuously changing working environment*.		
- # of layers of complexity already identified for processes. *Multiple instances of processes being suitable for different scope or complexity levels in project could enhance usability* - # of scenarios where the standard process had to be re-customized to be suited to a special project need: *this can be minimized by very robust or flexible processes*	(Fischermanns, 2009) [I]	V E

5.2.3 Information Technology

High level of visualization and user-friendliness: *IT-tools should be easy to handle and operate as well as being capable of providing all the information needed. These tools shall contribute to the efficiency and effectiveness of users operations.*

- Ø time it takes for a user to complete a specific task using the IT tools – *If applicable, ratio of this time to the manual process is also valuable* - IT-happiness-score: *measuring the satisfaction level of stakeholders with the CM IT system* - % of modules with Poka Yoke (mistake proofing) to prohibit the users from common mistakes	(Nielson, 1993) (Nielson, 1993) (Heinecke, 2012; Plaisant, 2010; Schneiderman, 1996)	V V U
Integration of CM-tools with other IT-Systems		
- # of solutions used to satisfy the needs of CM activities throughout the lifecycle: *The lower this number, the better consistency of information. When several user-developed tools need to be used, integration is not successful.*	[I]	V
CM-Functionalities and processes covered by IT tools – *Commercial PLM soluitions could stick to standard functions to match all usages rather than covering all needs of different discipline and environments.*		
- # processes modified due to the limitations IT-systems*: shall be minimized because the IT-systems must be specified by CM needs and processes and not the other way around* - % of CM-activities that are supported by a module in the solution - The ratio of used modules over the total available modules: *This could illustrate the opportunity of enhancing efficiency by utilizing already available functionality of IT solutions*	[I] (Lyon, 1999; Watts, 2008) [I]	E V E
Consistent and standard IT-Tools for all Lifecycle Phases in the entire organization: *Consistency of tools that are used for each purpose throughout the organization, could facilitate knowledge sharing and common language.*		
- # cases during the product lifecycle where the format of information needs to be changed because of different IT-tools or databases - # of tools used for a specific purpose in the entire organization	(Lyon, 1999) (P.C. Patton, 2007)	V V
Authorization Capabilities for different CM activities		

- # of CM processes that could individually be restricted in access management - Frequency of access right reviews for all employees	(Benantar, 2006; Lampson, 1971) [I]	U E

5.2.4 Organization & Value Stream

Suitable CM Organization Structure with Respect to Organization Complexity		
- Ratio of CM resources over project complexity: *Depending on the project scope, complexity and lifecycle duration, the amount of resources alliocated to CM practices shall be determined efficiently* - Ratio of centralized resources over project-based resources: *for managing various complexity levels, different degrees of central integration, defined as communication, control and coordination, is necessary*	(Hass, 2003) (Baccarini, 1996) (Baccarini, 1996)	U U
Defined Roles and Responsibilities for CM Personnel: *all required roles need to be defined clearly and the responsibility needs to be distributed to individuals*		
- Ratio of conflicts due to unclear responsibilities over total number of employees - % roles with clear responsibility descriptions and work instructions	(Hass, 2003) (Rouse & Sage, 2011) [I]	U E
Cross-Functional Collaboration Among Different CM Stakeholders: *CM is a cross-functional discipline, which is why the collaboration between functions is required by CM's very nature*		
- Ratio of functions involved in CM processes over all functions in the project - # of instances where changes were not communicated to the right stakeholders	[I] (Eckert, et al., 2004)	E V
Consideration of Suppliers and Subcontractors in CM Activities: *suppliers need to be integrated in CM activities to ensure standard CM practices for outsourced functions*		
- # non-conformities resulted from lack of value-chain involvement: *if the exchange of information about changed specifications, requirements etc. does not work well, non-conformities can occur* - Ø training hours for the involved subcontractors	(Hass, 2003) (Watts, 2008) (Rouse & Sage, 2011)	U U

Involvement of Key Stakeholders in Major Configuration Changes		
- # of complains after implementation of changes - Ratio of successful changes over all changes: *Involvement of all relevant stakeholders in decision making process increases the chance of change to be successful*	(Rouse & Sage, 2011) [I]	V E

5.2.5 Knowledge and support

Standard CM Terminology and Knowledge Support Accessible by All Stakeholders		
- Frequency of updating the CM standards and training database: ensuring actuality of information - Ø time taken for employees to obtain knowledge on CM functionalities - # of different terms used for a specific feature throughout the organization: *due to a lack of standard terminology, there could be several different names used to refer to a specific item. This increases the confusion and problems with communication.*	(Maier, 2004) (Lehner, 2009) (Lehner, 2009) (Maier, 2004) [I]	V V E
Regular CM-Related Training Activities: *Employees shall be trained regularly to ensure the quality of work*		
- % of CM activities for which training is available - Ø amount of training taken by each employee associated with CM - # of mistakes due to lack of proper training per employee	(Watts, 2009) (Watts, 2008) [I]	V V E
Accessibility and Promotion of Latest Standards, Lessons Learned, Best Practices and Benchmarks		
- % of projects for which CM lessons learned is recorded and accessible by future stakeholders - % coverage of CM activities by standards and benchmark documents accessible to corporate search engines - # of instances where lessons learned used per each project month	(Maier, 2004) (Maier, 2004) (Hass, 2003)	V V U
Support and Empowerment of CM Discipline by Senior Management: To ensure the fulfillment and acceptance of CM, management support is necessary		

- % of key CM meetings in a project with senior management involved - % annual budget allocated to enhancing CM functionality and trainings	[I] [I]	E E
Communication of CM Benefits to Stakeholders by Senior Management: *For the purpose of motivation and acceptability of CM throughout the organization, there is a need for the senior management to regularly communicate the benefits to all stakeholders*		
- # of events were CM is promoted as an influential discipline by senior Management	(Doppler, 2011)	U

Due to the main focus of this chapter, only quantitiative KPIs that could be measured either internally or by surveys were developed. There are of course several potential qualitative measures that could be associated with each dimension that requires deep analysis and could only be possible by performing detailed audits. Therefore, such metrics were excluded from the course of this study but could supplement the model for audit purposes.

Addition of the KPIs is verified to enhance the model's functionality to not only assess the implementation and availability of several CM success factors in an organization but also measure the quality of such success factors and provide a baseline for continuous improvement of this quality level. A future research potential would be to reapply the model in an extended cross-industry assessment to provide deeper information on KPI ranges for each dimension with respect to size and industry sector of target organizations. This could act as a benchmark for future assessments and comparisons of any assessed organization with their industry sector. At the same time, this could lead to providing a better roadmap for evaluated organizations for further growth and reaching higher maturity levels. This is of course dependent on the applicability of the above KPIs to all organizations as well as possibility of extracting the suitable information represented by each KPI over an extended period of time to get meaningful metrics.

6. Decision Support System for Engineering Change Management

Empirical investigation on the application of ECM processes in various industries illustrated that one of the main challenges, especially in complex engineering environments, is to perform a comprehensive change impact analysis before decision making. As is the case in such environments, very often long lifecycles of the products or facilities and loss of information or knowledgeable team members could contribute to this challenge. Meanwhile, the involvement of several disciplines and departments could add to the challenge in the sense that considering various impacts as well as interactions of systems and sub-systems becomes tricky and impossible. A simple example was observed at CERN where modification of a subsystem for cryogenics led to limitations in accessibility of another subsystem for maintenance activities and this was only realized after the modification was approved and implemented.

One of the benefits of such environments is the availability of rich data and information that is collected throughout different phases of the long lifecycle. If this information especially in the context of engineering changes, could be well structured and utilized to generate useful knowledge, powerful tools could supplement the decision makers' analytical abilities by considering more variables, scenarios and experiences. Therefore, the main goal of this section is to provide a suitable representation of an information and process model for a Decision Support System in the field of Engineering Change Management in order for an improved Change Impact Analysis method.

In doing so, a Knowledge Base needs to be established to store and structure the data, information and knowledge accumulated within the organization. Furthermore, appropriate mechanisms for reasoning about the stored knowledge to draw conclusions for the cases in question need to be developed. In addition to that, a coherent information flow needs to be set in order to assure system functionality. The model proposed in this section shall also comply with the requirements set forth by real-life requirements and must genuinely support decision making in an ECM setting. In order to tackle this problem correctly and comprehensively, this research will be structured in the following format:

First, an overview of current ECM practices and identification of best practices in processes and IT tools proposed in the literature is provided. Furthermore, an analysis of Decision Support Systems within the field of Change Impact Analysis will be carried out and on the basis of these results, combined with basics on the individual elements of a Knowledge-Based Decision Support System, a suitable model for a Knowledge Base will be developed. In order to retrieve an estimation of the change impact a mechanism for retrieving similar cases from historical records will be presented. The results of the previous cases serve as a first rough estimation of the impact that the current case is likely to have. Next, Case-based Reasoning component of the Knowledge-Based system is introduced, which is responsible for further reasoning about the case results with adaptation rules and infers an impact estimation more suitable to the case at hand. At the end, the architecture of the system and the interplay of the introduced components is further characterized.

6.1 Engineering Change Management

One of the key capabilities of companies in the current, rapidly changing competitive environments is their ability to quickly innovate on a high level, more so on products which have already been released. Especially challenging factors are shorter product life cycles, faster technological changes, demanding and sophisticated customers and increasing globalization coupled with a convergence of industries (Wang & Kourouklis, 2013). Thus, an alteration to form, fit or function of a part, drawing or piece of software of any magnitude or at any point in a product's lifecycle is considered a change activity (Cross, 1989).

As there are various terms and differing definitions, a common understanding of what is considered EC must be established and a clear distinction from other terms within the field must be drawn. Furthermore, the EC process is nothing but a part of the overarching PLM approach and can only be understood in its entirety when placed in the proper context. Different requirements on EC have led to a variety of process models which incorporate a differing number of steps and phases (Jarratt, et al., 2010). Thus, a very important competitive factor for any company is the efficient and effective management of engineering changes, which must consequently be supported by the best tools possible.

To lay the foundation for further improvement activities on ECM processes, a detailed literature analysis of Engineering Change definitions, various types and triggers was performed and is thoroughly presented in Appendix B.

6.2 Decision Support for Change Impact Analysis

When designing a tool for the decision support in an organization, many factors must be considered. It is especially important to identify the suitable setup for the type of available data and the kind of decision in question. Therefore, an analyis of the various factors such as Knowledge management science, characteristics of knowledge bases as well as different types of decision support systems were performed and the extensive analysis is presented in details in Appendix C.

6.2.1 Knowledge-Base Architecture

Within the literature, there is an apparent lack of concrete approaches and examples on how to structure a Knowledge Base (KB). Nonetheless, there is a tendency to divide the KB into several layers. One example of this is contained in a study by Kohn et al., who divide the KB of their system for the handling of product models into the following layers (Kohn, et al., 2013):

Layer of the knowledge base	Content	Function	Output of information and guidelines
General model layer	Container of application-independent aspects of model definition and distinction	Represents abstraction of the application layer and stores general procedures and processes	General insights and rules, as well as best practices
Application layer	Contains specific models of products, processes and structures	Provides structure for the various product modeling situations and proposes steps in the product development process	Checklist and rules and guidelines for the handling of product models
Project-specific layer	Concrete cases of used product models and the complete documentation for cases	Allows for the search and reuse of existing product models and documents	Suggestion of rules and specific application advice

Table 4 – Knowledge Base Layers according to (Kohn, et al., 2013)

Another approach to structuring a Knowledge Base is in the case of Arian and Pheng's study of effective management of contract variations. In identifying, analyzing and managing the variations in contracts, they employ a KBDSS with the following KB structure (Arian & Pheng, 2006):

Layer	Content
Macro layer (level 1)	General data about all existing projects within the database
Micro layer (level 2)	Detailed variation within the bounds of a project
Effects / control layer (level 3)	Effects of variation and suggested solution for potential control

Table 5 – Knowledge Base Layers according to (Arian & Pheng, 2006)

With the insights contained in the previous examples, Niknam et al. have proposed a Knowledge-Base structure consisting of a „Facts Layer", a „Rules Layer" and a „Strategy Layer", which is organized in a hierarchichal fashion (Niknam, et al., 2014).

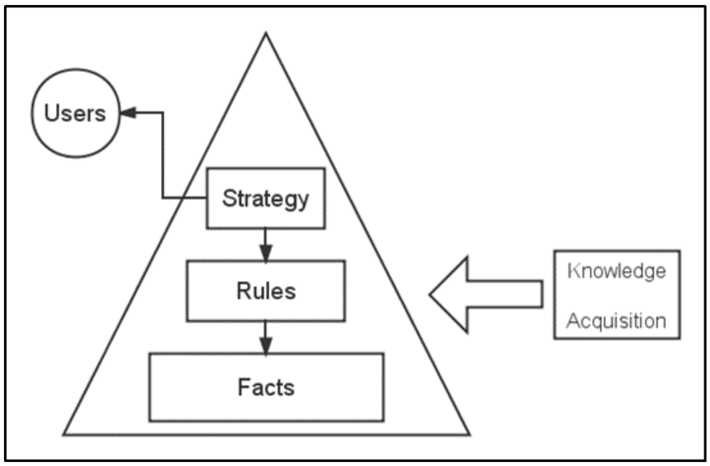

Figure 42 – Knowledge Base Structure – Based on (Niknam, et al., 2014)

Each of the layers serves a different function within the Knowledge-Base:

- The facts layer is the foundation of the proposed Knowledge-Base and contains all the raw data acquired by the system. One can, in accordance with the approach by Arian and Pheng, even divide this layer into a macro and a micro view (Arian & Pheng, 2006). As far as the acquisition of data is concerned, various sources are to be employed.
- Included in the rules layer is the knowledge deduced from the facts, which uses the case properties and relationships to establish links between the various data points and to find similarities between previous cases and the engineering change case currently at hand.
- Serving as a further level of abstraction, the strategy layer uses the similarities found by the rules layer to further refine the prediction quality. This is done by employing the knowledge acquired through expert input or machine learning techniques.

Supporting the decision makers for EC cases, this type of system must be included in the overall process of Engineering Change Management. It is especially important to clarify the integration in the workflow and the intended exchange of information between users, knowledge engineers and the Decision Support System.

6.2.2 Enhanced ECM Process with Impact Evaluation

During the impact evaluation of an Engineering Change case, numerous factors and interdependencies must be included. Within complex product structures, highly specialized production processes and large supply chain, a manual evaluation seems almost impossible. This is where the proposed Decision Support System can offer helpful guidance to decision makers.

In accordance with the previously proposed Knowledge-Base structure, the impact evaluation during Engineering Change Management based on the historical data is implemented by the following process (Niknam, et al., 2014):

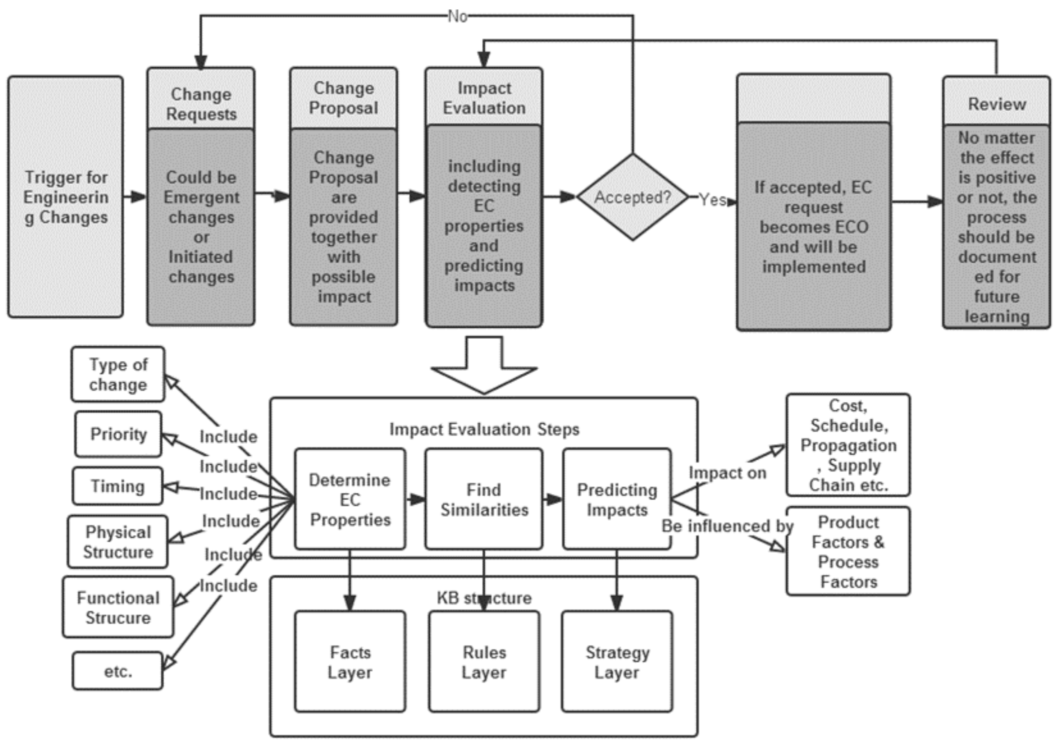

Figure 43 – Process Model of the KBDSS – From (Niknam, et al., 2014)

1. Initially, the EC case is input into the system. Based on the information model used for modeling the knowledge about change cases within the domain, the structure proposes various important aspects to be considered in the evaluation. The properties of the case are dependent on the case at hand and vary for each instance of a change initiative. Within the context of the Knowledge-Base, this step is carried out on the facts layer of the Knowledge-Base.
2. Using similarity measures, the relationships between the current case and previous cases within the Knowledge Base is determined on the level of the rules layer. The higher the similarity rating, the more relevant is a previous case for the solution of the Engineering Change instance currently under evaluation.
3. A set of most similar cases is retrieved from the case base and used for the subsequent steps. One can imagine a number of methods to select cases, such as a threshold similarity value, or the retrieval of a fixed number of the most highly ranked cases.

4. Predicting the impacts of the Engineering Change initiative is undertaken on the rules layer. Based on expert knowledge and rules derived from previous cases, an estimation of the effects of the change case is calculated.
5. In determining the feasibility of the proposed Engineering Change initiative, the decision maker reviews the output data from the Decision Support System. Depending on this judgement, the Engineering Change Request is either denied, returned for reiteration or accepted. A positive evaluation causes the request to develop into an Engineering Change Order and subsequent steps for the implementation are undertaken.
6. As the system is based on using the data, information and knowledge contained in previous cases, it is important to review and incorporate the feedback of a completed case. This way, the case base steadily grows and over time, the performance of the Decision Support System will optimize.

The interaction between the individual users and components of the system is illustrated in the following figure describing the information flow.

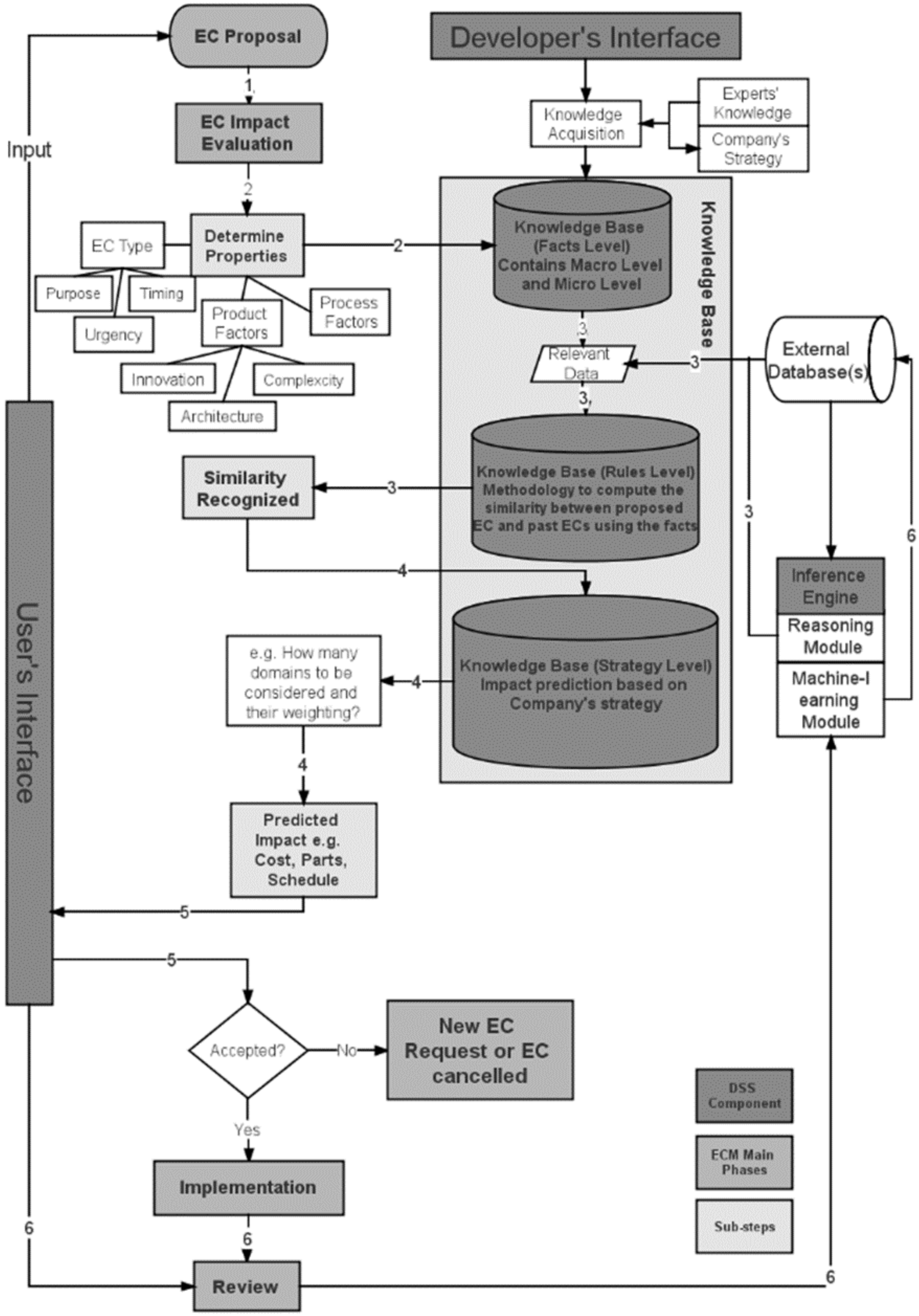

Figure 44 – Information Flow of the KBDSS – From (Niknam, et al., 2014)

With the suggested system, the efficiency and effectiveness of the Engineering Change process can be improved. This is particularly important when considering the discrepancy between the EC process maturity and the perceived development level of the tools supporting the decision makers (Niknam & Ovtcharova, 2013). Nonetheless, in order to make implementation possible, a more thorough definition of the various components and detailed workings of the system is required and illustrated in the following sections.

6.3 System Components requirement

Any Knowledge-Based System, such as the one described in the previous chapter, achieves its reasoning power through the explicit representation and use of different forms of knowledge about a field of application (Diaz-Agudo & Gonzalez-Calero, 2007). Yet, there are numerous ways of representing and using knowledge. Therefore, it is essential to provide detailed descriptions of the individual features of the knowledge represesentation and the problem solving approaches.

In doing so, one must be sure to include the various types of knowledge that are present within an organization. Apart from the explicit knowledge codified in rules and guidelines, tacit, descriptive, procedural and reasoning knowledge need to be considered in order to achieve maximum organizational agility and competitiveness. With the use of a Knowledge-Based Decision Support System, organizational learning is promoted, a more consistent procedure for impact assessment is introduced and, as a result, decision-making is improved (Noran, 2009).

Of the various kinds of Knowledge-Based Systems, the best fit for the domain of Engineering Change Management is identified as a hybrid reasoning system, based on Case Based Reasoning (CBR), which utilizes not only the knowledge contained in the previous cases, but also the domain knowledge gathered from experts in setting up the information model and by the explicit recording of domain rules and factor relevance.

Within the literature, the case-based approach is divided into the four phases (i) retrieve, (ii) reuse, (iii) revise and (iv) retain, which is referred to as the CBR cycle (Aamodt & Plaza, 1994).

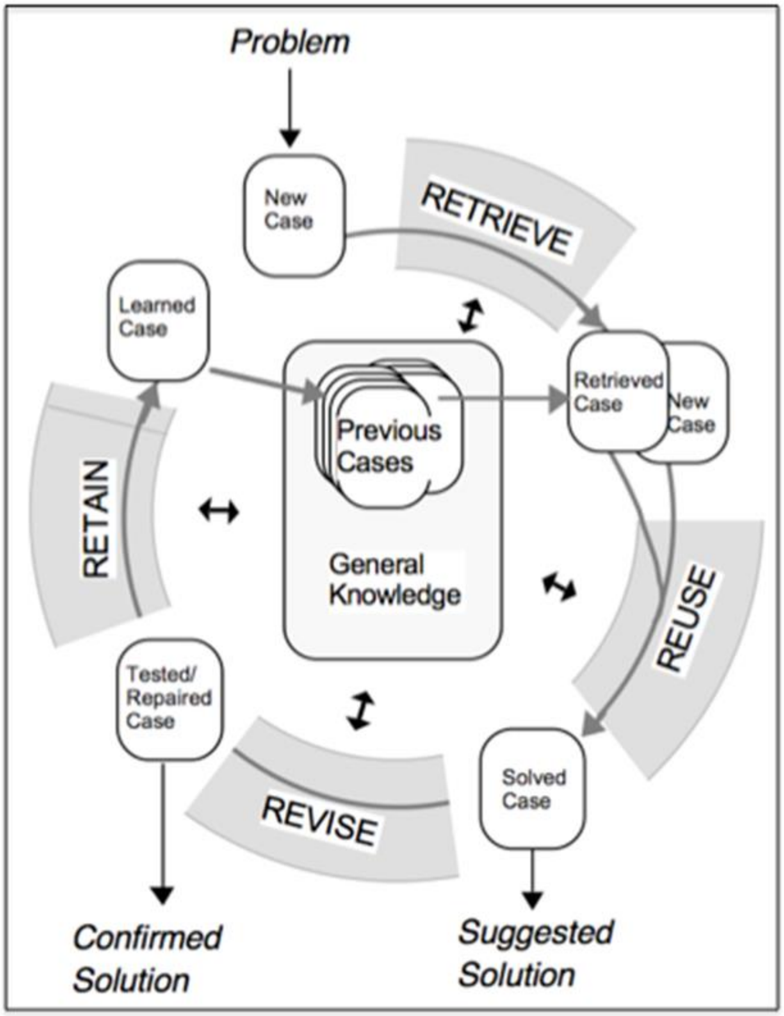

Figure 45 – The Case Based Reasoning Cycle (Aamodt & Plaza, 1994)

At the retrieval phase, the most relevant previous cases are obtained from the case base with the use of a similarity metric. Within the reuse phase, a preliminary solution for the new case is created using one or more retried cases. During the revise phase, validation of the proposed solution is undertaken, possibly with the help of an expert user. Finally, the retain phase incorporates the additionally gathered knowledge into the system (Prentazas & Hatzilygeroudis, 2007).

The introduced system follows much of the same approach, yet several questions are left to be addressed. Achieving an operational system requires the coherent definition of the domain knowledge, as well as of problem solving methods used for reasoning about the cases.

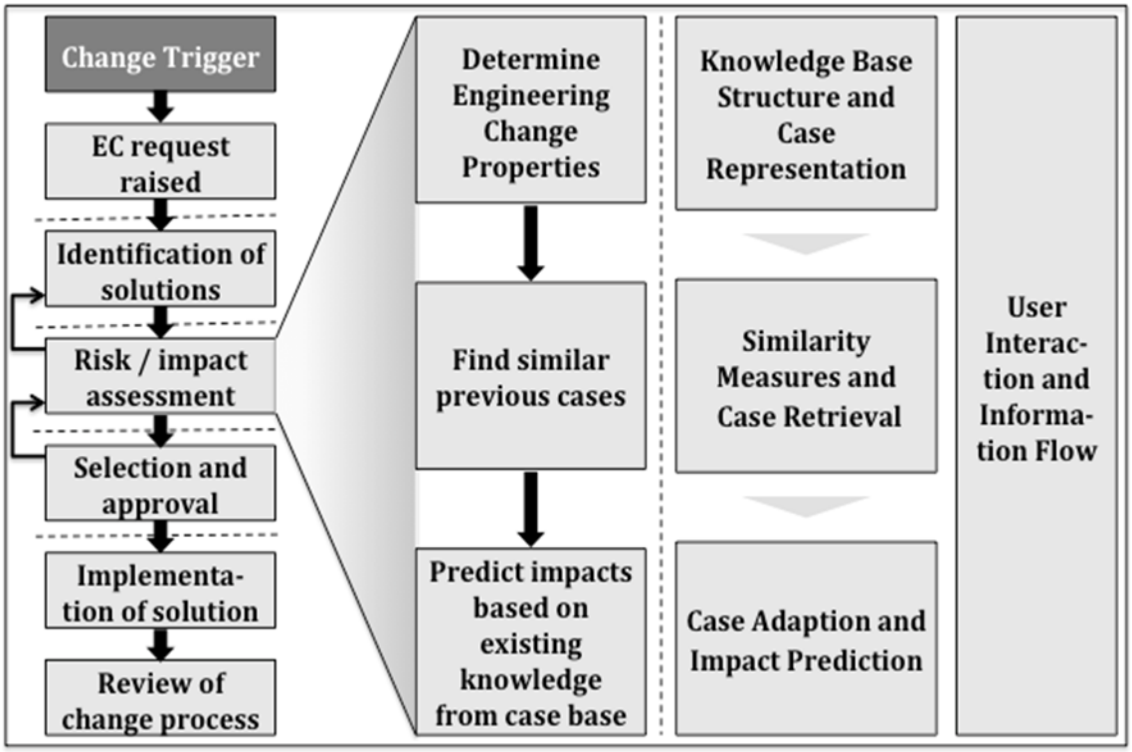

Figure 46 – Development Roadmap for the proposed System

Detailed structure of the Knowledge Base is extremely critical and includes the main part of the information model of such a system. Within this area, use of ontologies is promising, as they allow for an integration of domain knowledge already at the point of system design and greatly increase the capabilities of the system for reuse of knowledge and specific reasoning (Gomez Perez & Benjamins, 1999). In order to be able to apply the methods required for a Knowledge-Based Decision Support System, reliable approaches and algorithms must be proposed for the problem steps of similarity evaluation and impact prediction. Finally, a general architecture of such a system is to be further detailed and the information flow needs to be modeled.

6.3.1 Knowledge-Base Structure and Case Representation

Within the domain of Engineering Change Management, there is no standard Knowledge Base structure available. It is therefore essential for this work to define a structure, which includes all the relevant data points of an Engineering Change initiative, as well as the domain knowledge gathered. As ECM is part of the overall Product Lifecycle Management, only a data model congruent to the existing approaches within this field will be feasible for implementation. Therefore, Ontologies were used to construct the cases within the facts layer of the knowledge base. In addition to that, it is necessary to establish a consistent representation of the rules layer, which is responsible for the similarity assessment between the current and previous cases. Therefore, a model for comparing cases, based on the case ontology and a concept based similarity measure is established.

The most complex issue at hand is the definition of the strategy layer, which further refines the solutions of the retrieved similar cases to better match the current Engineering Change case. In order to achieve this, adaptation rules will be defined, not only including the antecedent and consequent parts, but also recording the frequency and confidence of the rule so that selection of the most appropriate adaptation rules becomes possible.

Through the coherent definition of the Knowledge Base elements in the context of ontologies, the system is able to integrate domain knowledge and consequently enables a more flexible and contextual reasoning and inference process. As a result, a better decision support can be achieved through this context sensitive approach. The detailed description of this section could be found in Appendix D.

6.3.2 Similarity Assessment and Case Retrieval

In trying to draw conclusions from previous Engineering Change cases, it is assumed that cases with similar determinant features result in comparable impacts. Thus, it is necessary to identify a reliable method for evaluating the similarity between a current case and the cases stored within the case base.

One of the most common approaches to similarity assessment is the k-nearest neighbor matching, which, as the name suggests, returns the k nearest neighbors of a query. This is achieved by applying a distance function on a d-dimensional vector space between the query and the set of points within the space (Wilke, et al., 1997) (Prentazas & Hatzilygeroudis, 2007).

However, this approach fails to sufficiently take domain knowledge into consideration (Patterson, et al., 1998). Especially within the domain of ECM, a rich repository of information is available within the database and the representation of cases with the help of ontologies provides the possiblity of including not only attribute, but also concept similarity within a given feature.

With the help of a similarity measure, a ranking between previous cases regarding their relevance for the current case is possible. Within this work, a number of cases is retrieved for further processing, while deriving the previously introduced Adaptation Knowledge Requirement, which serves as the input for the subsequent step of case adaptation. Detailed information on methods utilized for similarity assessment and case retrieval could be found in Appendix E.

6.3.3 Adaptation Rule Generation and Application

The framework of Case-Based Reasoning aims at reducing the effort of knowledge acquisition, called the knowledge acquisition bottleneck, related to Rule Based Systems. Yet, many of the real-world applications are mostly case retrieval systems, which do not truly take on the task of reasoning. This is due to the fact that it is difficult to obtain high quality adaptation knowledge and thus the potential benefit of CBR in reducing the knowledge engineering load is even aggrevated as one needs to gather both case knowledge and adaptation knowledge (Patterson, et al., 1998).

Nonetheless, a number of systems have emerged, using a wide variety of methods for case adaptation, ranging from a simple substitution of one component to approaches involving the modification of the overall structure of the solution (Mitra & Basak, 2005). There are several reasons why the inclusion of adaptation rules is essential for a high performing system. In fact, the two approaches of an instance based learning from previous cases and the inclusion of rule-induction paradigms compliment each other (Hanney & Keane, 1997).

The use of adaptation knowledge provides a far wider and better reach of a system, as the problem at hand need not entirely match any case recorded within the case base. Furthermore, the previous cases already provide an approximate solution, which is grounded in actual experience and needs only be adapted to the current case (Hanney & Keane, 1997). Therefore, the hybrid approach of instance-based and rule-induction paradigms yields great benefits and is the cornerstone of this work. In Appendix E, several approaches for adaptation are introduced, before several sources for adaptation knowledge are discussed, including an algorithm for the automatic retrieval of rules from the case base.

6.4 System Architecture and Information Flow

At the core of the system is the task of providing guidance to a decision maker within the Engineering Change Impact Evaluation step. In doing so, the case currently under evaluation is described through the input of key factors into the system. Based on previous similar cases and adaptation knowledge, an estimation of the impacts the current change has on cost and effort measures is determined.

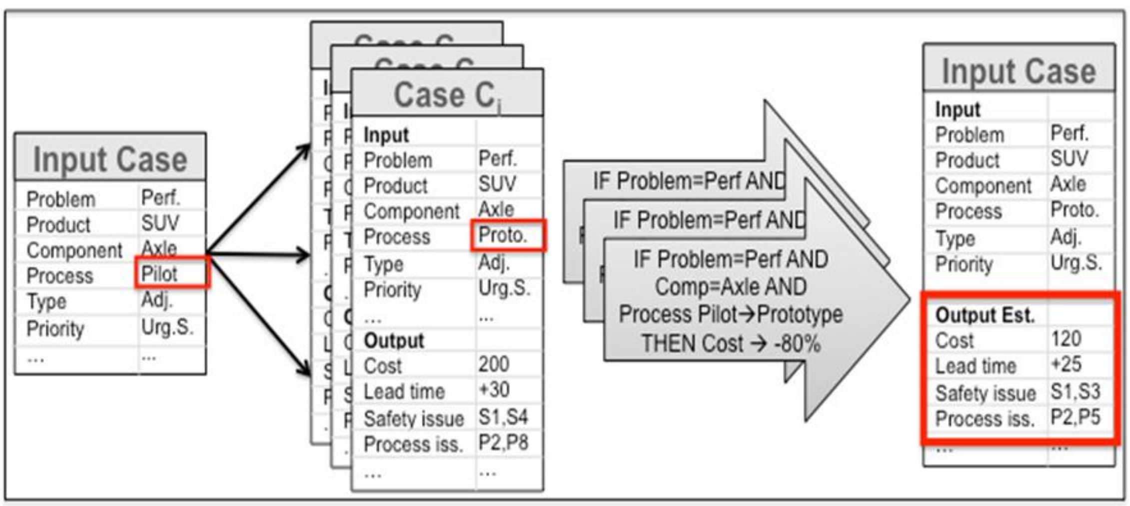

Figure 47 – Basic Functionality of the proposed System

In doing so, many different parts of the system need to be coherently defined and interact seamlessly to provide a consistent impact evaluation. In the following paragraph, the architecture of the proposed system is introduced, before guidelines on implementation are given and the chapter is concluded with a detailed description of the information flow between the individual components of the system and the user.

6.4.1 System Architecture

One of the principles in developing this Knowledge-Based Decision Support System is the separation of the domain knowledge from the Problem Solving Methods. This is essential for the reuse of the components in dynamically changing IT environments. It is therefore possible, based on the clear definition of the information model, to reuse the knowledge stored in the Knowledge Base with a range of different algorithms to solve various kinds of problems.

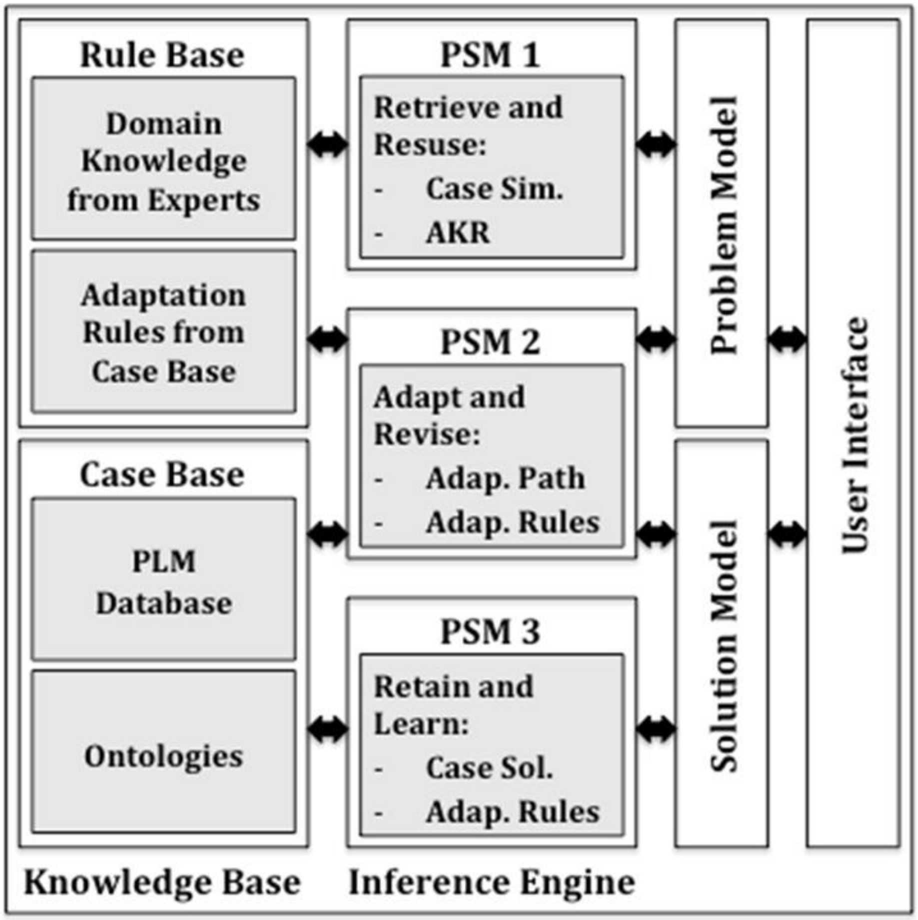

Figure 48 – System Architecture

At the core of the system is the Knowledge Base, which is comprised of the Rule Base and the Case Base. Within the KB, the knowledge about Change Management is integrated in a number of ways. The most basic is the representation of the various data points accumulated in the PLM database. This data is further enriched by setting up ontologies for most, if not all, of the features and thereby providing an enhanced representation of the cases, to form the Case Base of previously implemented change initiatives.

The Rule Base is the second component of the Knowledge Base and includes, as the name suggests, the adaptation rules for aligning the solutions of the retrieved similar previous cases to the situation of the current Engineering Change initative. In order to retain the domain knowledge from experts, rules about factor relevance and certain dependencies are recorded. This is an essential step for a high performing system, however a complete coverage of all possible adaptation requirements would not be feasible due to the high effort required. Therefore, these rules only provide the basis and a guideline for the automatic retrieval of adaptation rules from the Case Base.

The second major component of the system is the Inference Engine, which consists of the two main Problem Solving Methods. These are responsible for case retrieval and reuse based on overall similarity, as well as for case adaptation and revision. Within the proposed system, similarity is determined by weighing the differences in features and concepts between two case descriptions. Based on the obtained rating, a number of similar cases is retrieved. Employing the second Problem Solving Method, the retrieved cases are adapted to match the situation of the current case, before the decision maker is presented the results to guide the choice of subsequent action. The foundation for the successful case adaptation is a Rule Base populated with rules of high quality, so that the retrieved cases can be properly matched to the current situation.

Another Problem Solving Method is responsible for learning and retaining knowledge. This is where the experts and knowledge engineers are able to input the domain knowledge into the rule base. Furthermore, this component is responsible for retaining the solutions for a completed change initiative and for generating the adaptation rules resulting from the added information.

Finally, the User Interface allows the decision makers, as well as the domain experts, to interact with the system. With the help of the problem model and the solution model of the system, the users are able to input and receive information in a structured way. In the proposed setup, there is no distinction between a developer's interface and an interface for users. This is due to the fact that upon completion of a change initiative, not only the knowledge engineers, but also the decision makers can provide valuable input to the Knowledge Base.

Having introduced all the components constituting this system, the next section will introduce frameworks and tasks for the implementation of the proposed Knowledge-Based Decision Support System. It is important to note at this point, that the implementation is highly dependent on the domain. Furthermore, the success of such a system is not only based on the realization of a complete and coherent application, but also on the quality of the provided data.

6.4.2 Implementation Framework

There are several frameworks mentioned in the literature which can provide the basis for implementing the proposed system. One of them is myCBR, which is an open-source tool for developing Case Based Reasoning system and is mostly intended for prototyping applications focusing on the similaity assessment and retrieval phases (Stahl & Roth-Berghofer, 2008).

Another framework, called IUCBRF, was developed by a group at the Indiana University with the goal of providing an open-source solution for academic CBR projects (Bogaerts & Leake, 2013). This framework, as many others, does not feature sufficient functionality and has not been developed further.

Within the data mining community, Weka has been established as a central collection of machine-learning and data mining resource and is considered an open-source reference tool for many academic applications (Hall, et al., 2009). However, the problem-solving methods and object-based data structures of the proposed system are not easily implemented within Weka.

One of the most popular frameworks for implementing Case Based Reasoning systems is jCOLIBRI, which is developed by a team in Madrid, Spain (GAIA, 2013) . Within the current version jCOLIBRI2, components for supporting the entire

CBR cycle are included. It is designed to facilitate the construction of different configurations of reusable methods and a software composition tool allowing for an easy implementation (Recio-Garcia, et al., 2014).

Even though the jCOLIBRI2 framework does not currently implement the kind of complex Problem Solving Methods proposed for this system, it does provide the opportunity to include ontologies within the structure (Recio-Garcia, et al., 2007).

It is therefore suggested to use the jCOLIBRI2 framework as a basis for implementing the proposed Knowledge-Based Decision Support System and to code the required methods for case retrieval, adaptation rule generation and adaptation rule application.

At the time of implementation, several tasks are to be completed before the system can be considered operational:

1. Definition of the case structure: Based on the data available within the PLM system, as well as an ontological description of the case features, both the case description and case solution model are to be defined.
2. Implementation of similarity function: Besides determining which features to include in the similarity assessment, the feature weights need to be set and the number of cases to be retrieved is to be determined.
3. Population of the Rule Base: With the help of domain experts, as many rules as possible are to be input into the system. These will guide the automatic retrieval of adaptation rules through case comparison of previous change instances from the Case Base.
4. Method implementation: Specifically the algorithms for similarity assessment, adaptation rule generation, and adaptation rule application need to be implemented so that they can be applied to the cases stored in the Case Base.

On the basis of the introduced components of the proposed Knowledge-Based Decision Support System and their implementation, it is still necessary to describe the information flow during the impact assessment in more detail.

6.4.3 Information Flow

Within organizations, the need for changing a part or component can arise due to a number of different factors. These change triggers initiate the Engineering Change process and an Engineering Change Request is submitted.

On the basis of the request, possible solutions are explored and recorded in an Engineering Change Proposal, which is subsequently analyzed in the Change Impact Evaluation step. This is where the proposed Knowledge-Based Decision Support System provides helpful guidance.

Through the User Interface, the initiator of the Change Proposal can enter the description of the proposed Engineering Change case. This information is then processed within the system, before it is presented to the decision maker for the Change Impact Evaluation. This approach can easily be implemented in the existing EC workflow process already established in many of the Product Lifecycle Management tools.

Proceeding from the input of the case description, the proposed system compares the current change initiative with the cases stored in the Case Base and retrieves a small number of the most similar cases for further processing. This is done using the ontological descriptions of the different features and assessing the concept similarity, as well as determining the feature similarity. These are combined to an Overall Similarity Measure using the feature weights.

The retrieved cases provide an approximation to what the impact of the proposed change initiative could be. In order to further refine the solutions to better match the current case, adaptation knowledge is used to adapt the case solutions. In so doing, adaptation rules, which are based on expert input, as well as the comparison of previous cases, are applied to the retrieved cases and an improved approximation of the change impact is derived.

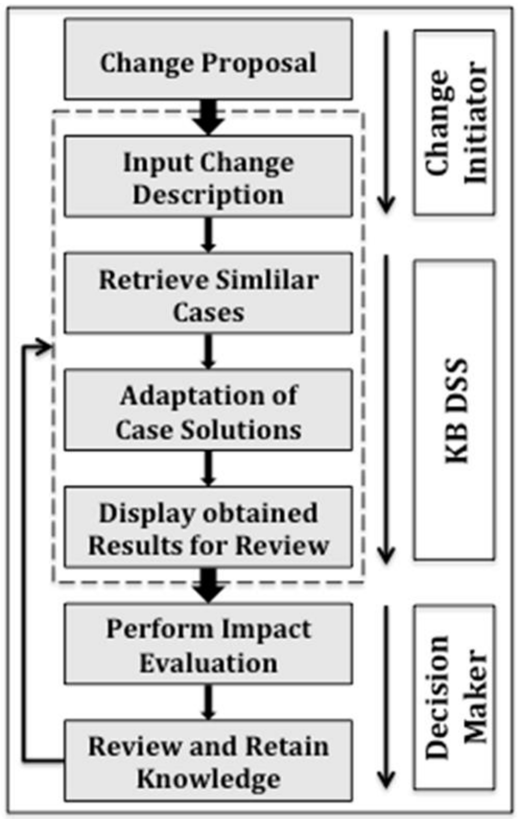

Figure 49 – Information Flow

The results determined by the system are then displayed to the decision maker. Within the domain of ECM, it seems necessary to base the decision on more than one retrieved and adapted case. That is why the author suggests the display of a fixed number of the most similar cases. Furthermore, for the sake of transparency, the decision maker needs to be able to understand how the estimations of the system are derived. Therefore, the similarity between the retrieved cases and the current change initiative are to be displayed. Additionaly, the adaptation path, as well as the confidence level of the individual adaptation rules need to be included in the information provided to the decision maker.

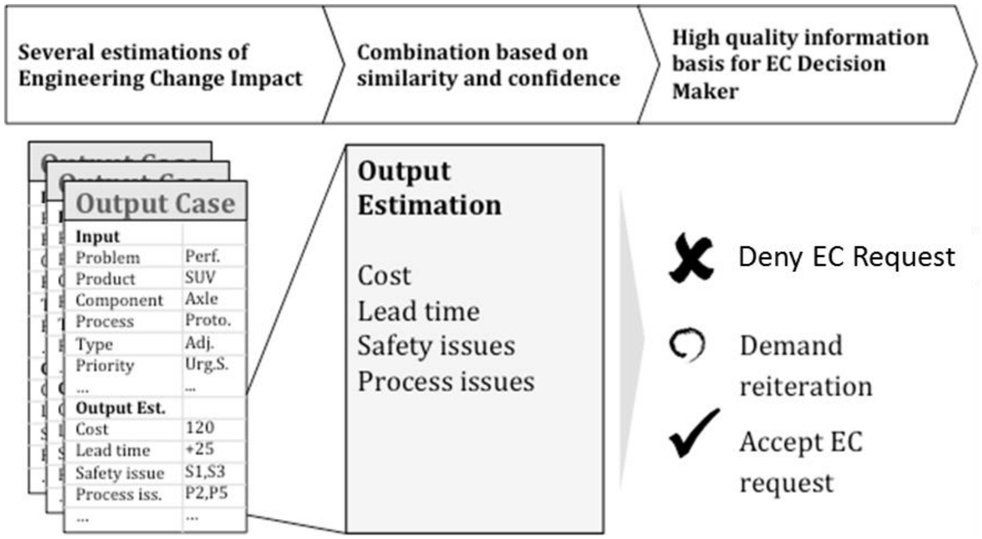

Figure 50 – Basis and Results of Change Impact Evaluation

It is then the task of the decision maker to utilize the data provided by the Knowledge-Based Decision Support System to compare the estimations on the impact of the proposed Engineering Change initiative with the potential benefits. Depending on the judgement of the decision maker, a range of actions are possible. Again, in line with the workflow established in most PLM systems, an Engineering Change proposal can be denied, sent back for reiteration, or be accepted.

While a denied request results in the conclusion of the particular change initiative, a demand for reiteration leads to a notice to the initiators of the Change Proposal to implement the alterations indicated by the decision maker and to re-enter the adjusted proposition regarding the change. Should a Change Proposal be accepted, an Engineering Change Order is issued and the implementation is scheduled.

The final step of the process is the review and evaluation of the initiated change. It is at this point that the knowledge acquired in the course of the change initiative is retained in the form of the case and its actual solution in the Case Base and also in the form of adaptation knowledge in the from of rules derived from the experiences gathered within the change initiative.

In order for the proposed system to provide valuable insight, a coherent structure and a tight integration of the individual components is required. Furthermore, the information flow must be properly managed, so that, based on the knowledge stored in the Knowledge Base, this Knowledge-Based Decision Support System can perform the reasoning required to provide a high quality estimation of the impact of a proposed Engineering Change initiative.

In the course of the preceding work, the components of the proposed system are thoroughly defined and the setup and interaction of the Knowledge Base and the Problem Solving Methods are specified in detail. The following section will provide insights on the potential and limitations of the proposed system and also gives an outlook on potential challenges.

6.5 Model Reflection

The proposed system attempts to support the decision maker by providing an estimation on the potential consequences of implementing a given change initiative based on knowledge that is already available within the organization. An overview of the complete model is illustrated in the following figure.

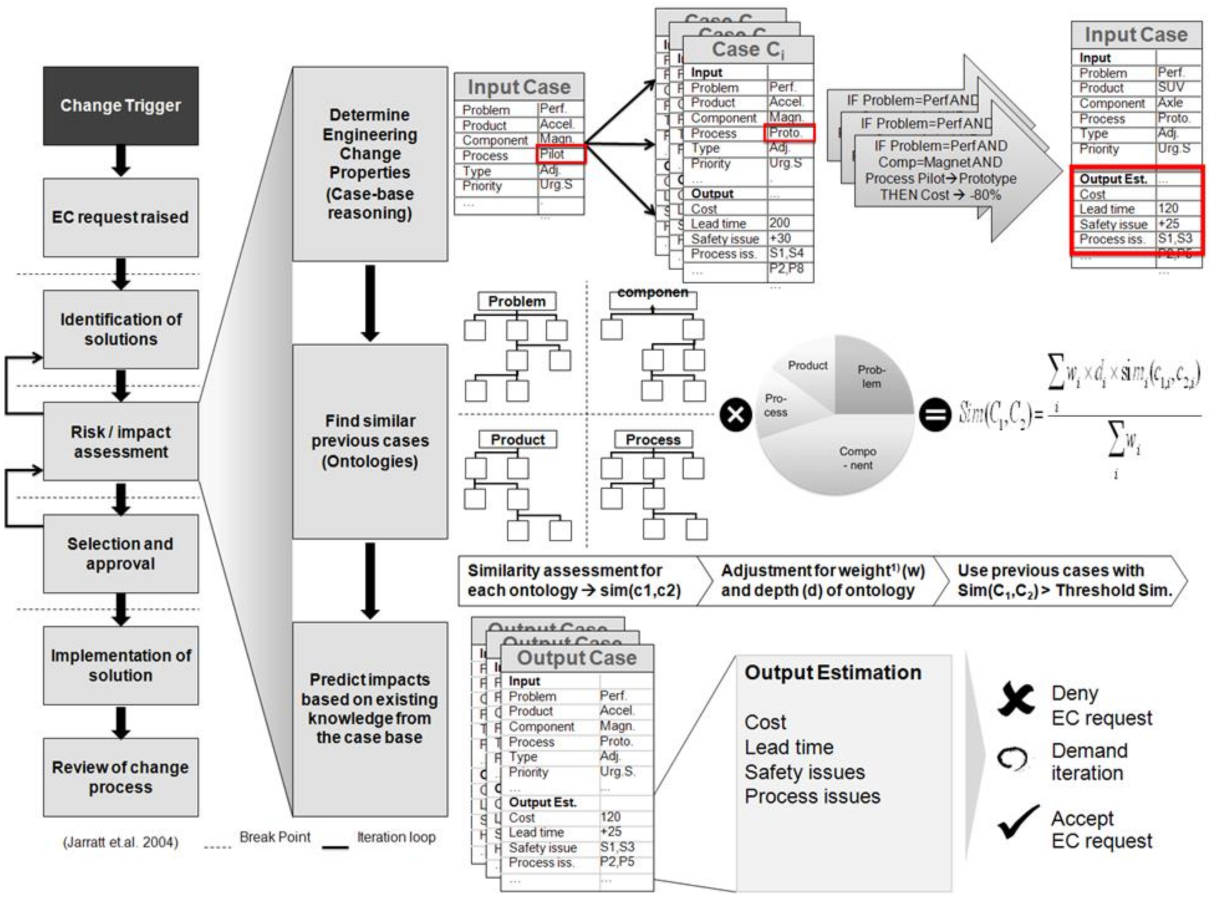

Figure 51 - Decision Support System for Engineering Change Management

Accordingly, one of the most essential prerequisites of a well functioning system is to have access to a comprehensive Knowledge Base populated with high quality data. Within the context of large organizations, one can assume that many change initiatives have been undertaken and recorded within the PLM system. Nonetheless, it is not guaranteed that the data pertaining to an Engineering Change case is available. Within this work, however, it has been shown that it is possible to derive or implement many of the features in a STEP compliant way and thereby ensure the integration in existing IT infrastructure.

The representation of the features of Engineering Change cases with ontologies provides valuable knowledge about the structures and interdependencies of the individual features. Moreover, it allows for a similarity assessment based on domain expertise codified in the form of the ontolgies, which also provide a uniform vocabulary and hierarchy for describing change initiatives within organizaitons.

While requiring a substantial amount of work, employing the Analytic Hierarchy Process for determining the feature weights in similarity assessment enables an organization to adjust the system in the future, as new features can flexibly be included and the weight can be adjusted to concur with current developments.

It is important to note that keeping the system up to date and inclusive of all the relevant data, information, and knowledge that is present in the organization is essential to the continual effectiveness in estimating change impact. Especially important is the inclusion of cases into the Case Base upon completion. Also, the people involved in the change initiative need to include into the Rule Base not only knowledge gained in the current change initiative, but all their acquired knowledge. Therefore, it is essential that the review and retaining step of the process becomes fully integrated into the workflow associated with Engineering Change Management.

Despite the ability of the proposed system to acquire adaptation rules from the Case Base, better results are to be expected when including domain knowledge in the form of rules about factor relevance and existing relationships between cases. Therefore, it is essential for improved system performance, to include as many domain experts as possible in the population of the Rule Base and to encourage the continuous update of knowledge.

Both of the main Problem Solving Methods, one for the retrieval and reuse of cases based on similarity, and the other for adapting and revising the estimation, are designed to make the results transparent to the decision maker, while maintaining a high level of accuracy. Within the literature, a large number of more complex mechanisms are described, but they lack traceability and are highly dependent on a comprehensive Case Base. Nonetheless, the independent structure of the Knowledge Base allows for the future evolution of the system methods. It is also important to maintain a coherent integration of the individual components of the proposed Knowledge-Based Decision Support System in order to ensure a high quality estimation of the impact of a current Engineering Change initiative.

The next steps for a successful implementation of the proposed system are all dependent on finding a collaborative organization that is willing to provide their data and domain expertise. On the basis of the definitions proposed in this work, the Knowledge Base needs to be populated with cases, ontologies and rules. Furthermore, the Problem Solving Methods for similarity assessment and case adaptation need to be implemented using the formulas and algorithms provided.

Developing a suitable User Interface is an open issue which needs to be addressed once the domain is chosen and the relevant factors for assessing change impact are properly defined for the specific organization. In the process of obtaining the Impact Estimations for the decision makers, transparency is essential.

For the implementation and consequent practical validation of the proposed system, the cooperation with a Product Lifecycle Management software firm appears especially advantageous. The existing data and workflows can be used to fully integrate this Knowledge-Based Decision Support System in the existing procedures established to handle Engineering Changes.

In the course of this work, based on the proposed information model, a Knowledge Base was structured and algorithms for inference were proposed. These components, together with a coherent information flow, provide the theoretical framework for an enhanced change impact evaluation process.

7. Conclusion

The challenge in the way of successful Configuration Management implementation in complex engineering environments derives from the complexity of systems and subsystems, level of technological advancement, product structure, interaction among components, number of involved disciplines, long lifecycle of products or facilities and the inherent risk in decision making. Even though the reason behind formulation of CM discipline in the first place was to handle complexity by keeping the consistency of products and information, its proper organization-wide implementation can sometimes appear time consuming and trivial to many organizations. This happens due to the lack of correlation between organization success and CM implementation. The optimum objective of this thesis was to emphasize on the importance of this discipline and contribute to the enhancement of Configuration and Change Management activities of complex engineering environments.

In order to achieve this goal, the first step was to identify the current overall status of CM activities in various industries to be able to pinpoint the gaps. This could only be done by the use of a Maturity Model as a tool to provide a comprehensive understanding of various aspects of Configuration Management in any target organization. Since, no Maturity Models for CM discipline was developed, with the help of literature, standards and practitioners in the field, the Configuration Management Maturity Model was developed. This model is utilizing several Critical Success Factors of the discipline as categories to be evaluated in target organization. In order to validate the functionality of the model, a Maturity Model Comparison Framework was also developed. Utilizing this framework, the CM Maturity Model was structurally compared to nine other validated models in the field and the results presented sufficient robustness of this model and its readiness for field testing.

After establishing the structural validity, the content and functionality of the Maturity Model was to be validated. This was achieved by a cross-industry maturity assessment project where the maturity of over sixty seven organizations across Europe were evaluated using the developed model. In addition to the usability validation of the model, strong and diverse information was gathered about the current status of Configuration Management implementation in various industry sectors with different sizes. The results of the two approaches provided the research with enhancement opportunities for the model as well as overall gaps in the CM implementation in various industries. Further research confirmed the two critical gaps as being the lack of Configuration Management Key Performance Indicators and process gap in Engineering Change Impact Analysis.

The absence of Key Performance Indicators was one of the main reasons behind the lack of understanding of the importance behind implementation of CM discipline. CM as a support discipline, is usually considered as a controlling and redundant discipline with no tangible contribution to the success and profitability of the organizations. Existence of a strategic objective that is driven by corporate vision and measuring the key metrics on a regular basis to move towards that strategic objective, could ensure the successful implementation of CM discipline and at the same time provide an understanding of measureable benefits associated with this discipline.

The process gap in Engineering Change Impact analysis is driven by the inherent complexity of the target organizations. As complexity increases, proper evaluation of the potential impact introduced by any change in systems or subsystems, requires detailed analysis of the change and involvement of several stakeholders. As the nature of complex engineering environments entails, high technological levels and number of critical assets could make this process very difficult and less accurate.

Realizing the main challenges, further research was performed to tackle the two problems and propose solutions for each. A set of KPIs was developed for each of the identified CSFs in Configuration Management and verified with industry experts. Utilization of such KPIs in the CM Maturity Model could enhance the evaluation methodology and provide measureable metrics for the maturity level of each assessed organization.

At the same time, an extensive research was held in the area of Engineering Change Management and a conceptual solution as a Decision Support System for Engineering Change Management was proposed. This was done by using the long lifecycle feature of complex engineering environments where there is a wealth of knowledge available from various phases of the project or previous similar projects. The information about changes that happened in the past, their characteristics and their impacts can act as a significant resource for decision makers in anticipating the impact for upcoming changes more accurately and enhancing the change impact analysis methodology. Therefore, the proposed Decision Support System was designed in a way to capture the main characteristics of any upcoming change, compare them against the characteristics of all historical changes and find the most similar change in the past. Based on that and the differences in the characteristics, the system would adjust the impacts of the old change and propose the new estimated impacts for the current change. This will assist the decision makers to consider various factors in their decision making and utilize the wealth of historical information for their benefit.

Due to the broad focus area in this research and the limited scope of the thesis, during the course of this work concentration was dedicated to the most important findings and research opportunities. As it is the case with many process improvement research works, the findings of this thesis could very well be enhanced and further developed by future deeper research, solution development and implementation of proposed concepts. The widespread application of this research subject, provides an opportunity for future utilization and adaptation in both research and industry.

The proposed CM Maturity Model was enhanced with measureable metrics and could be utilized towards deeper understanding of Configuration Management Maturity in various organization types. Implementation and utilization of this model in various industries could lead to “comparative” abilities of this model where assessments could be done per industry, organization size or geographical area.

The Maturity Model Comparison Framework can potentially be enhanced with metrics and provide scoring system for maturity models where models from various disciplines could neutrally be compared and enhancement opportunities for them could be recognized and benchmarked from other models.

The Decision Support System for Engineering Change Management, requires an implementation of proposed algorithms in a complex engineering environment where information about changes are captures and utilized to make a strong knowledge base. By collaborating with professional practitioners and decision makers, the model could be trained and enhanced to reach maximum accuracy in predicting the impact of any proposed change and can act as a reliable source for decision making support.

Appendix A. CM Maturity Assessment Survey Design

Configuration Management Maturity Assessment Survey is part of a collaborative project between Institute of Information Management in Engineering (IMI) at Karlsruhe Institute for Technology (KIT) and P3 Ingenieurgesellschaft.

The objective is to analyze the overall maturity of various industry segments with respect to their Configuration Management activities in order to find out the focus areas for further improvements.

The results will be published within the next three months as a booklet and as promised, you as a participant will receive a free copy of the published result.

<u>It is worth mentioning that the information provided here by you will be treated confidentially and no names will be used in the published results.</u>

The survey will take about 20-25 minutes and at the end there will be a section for your comments about the survey itself.

Thank you very much for participating in this survey

Section A: General

A1. Does your organization as a whole or some entities of your organization (units, projects) implement Configuration Management practices?

Yes, whole organization ☐

Yes, some entities ☐

No ☐

A2. Has your organization measured its Configuration Management Maturity level?

Yes, and according to a formal Configuration Management maturity assessment scheme ☐

Yes, but by means of an informal benchmarking towards competitors ☐

No ☐

A3. What Configuration Management maturity model scheme has your nization orgaused?

A4. Does your organization perform Configuration Management activities for both software and hardware items?

Yes ☐

No, only hardware ☐

No, only software ☐

☐

None

A5. How are the CM practices applied along the product/facility lifecycle phases of your organization? (Please check as many lifecycle phases as applicable)

Planning ☐

Design ☐

Manufacturing ☐

Operations ☐

Decommissioning ☐

A6. At a glance, how would you describe the level of maturity of your organization in matter of Configuration Management?

Initial, Only some individuals/groups might autonomously implement a few aspects of Configuration Management on their own, but there is no formal CM discipline in place. ☐

<p> Managed, there is no generic formal CM discipline in place, although some specific formal methods and processes may be in use for some projects. CM application depends heavily on the skills of the project or operations teams and of the availability of external support.</p> ☐

Standard, There are generic processes applied to most/all projects or operations activities. The formal processes are incorporated into the quality system of all projects. There is a limited need for external support. ☐

Optimizing, “Total Configuration Management” is permeating the entire organization. The Configuration Management activities and processes are regularly refreshed and updated. The organization has processes for collecting feedback to improve its CM-related activities and processes continuously. ☐

Section B: Strategy and Performance

Initial, Only some individuals/groups might autonomously consider a few aspects

Managed, no generic formal approach in place but some aspects might be applied in some projects

Standard, generic formal approach in place and most aspects are applied to most/all projects or operations activities

Optimizing,all aspects are applied to the entire organization and continuously being updated and improving

In the spirit the four possible levels of maturity shown above, how would you characterize?

B1. The Configuration Management Strategy and Policy of your organization:

Initial ☐

Managed ☐

Standard ☐

Optimized ☐

B2. The deployment of CM strategy among different organizational levels and units:

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

B3. The communication of Configuration Management Strategy and Policy throughout stakeholders in your organization:

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

B4. The KPIs (i) in your organization for measuring the performance of Configuration Management practices:

KPIs or Key Performance Indicators are defined measures to periodically assess the performance of organizations activities with respect to the defined strategy and objectives of the organization.

Initial ☐

Managed ☐

Standard ☐

Optimized ☐

Section C: Processes

Initial, Only some individuals/groups might autonomously consider a few aspects

Managed, no generic formal approach in place but some aspects might be applied in some projects

Standard, generic formal approach in place and most aspects are applied to most/all projects or operations activities

Optimizing,all aspects are applied to the entire organization and continuously being updated and improving

In the spirit the four possible levels of maturity shown above, how would you characterize?

C1. The Overall Configuration Management processes of your organization:

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

C2. The overall consideration of subcontractors in various CM processes of your organization:

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

C3. The specific processes of your organization for identifying and defining *configuration items (CIs)* (i)*:*

Configuration Items (CIs): A product or an aggregation of products that accomplishes an end-use function and requires separate identification. An item is designated as a CI for purposes of additional configuration management focus due to its complexity, logistic support requirements, acquisition strategy, or because it is intended to undergo configuration status accounting or verification and audit separately from other items.

Configuration items are end items or major components of end items, which typically have performance requirements allocated to them and documented in their own specification.

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

C4. The specific processes of your organization for managing product structure through all lifecycle phases:

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

C5. The specific processes of your organization for creating and updating different *(Functional, Physical and Allocated) baselines* (i):

Baseline: defined and approved collection of configuration documentation (e.g., specifications, drawings, etc.) established at a specific point in time. The baseline is a formal, controlled and maintained set of data that serves as a basis for defining change.

Functional Baseline: The approved functional requirements of a product or system, describing the functional, performance, interoperability, interface and verification requirements, established at a specific point in time.

Allocated Baseline: The approved requirements of a product, subsystem or component, describing the functional, performance, interoperability, interface and verification requirements, that are allocated from higher-level requirements, as established at a specific point in time.

Product Baseline: The approved product configuration documentation of a product or system established at a specific point in time.

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

C6. The specific processes of your organization for managing *Engineering Changes, i.e. ECPs and ECOs* (i):

ECP (Engineering Change Proposal) is a document used to describe, justify and submit a proposed engineering change. ECO (Engineering Change Order) is a document used to order a change based on an analysed and approved ECP.

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

C7. The specific processes of your organization for Change impact analysis and change classification:

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

C8. The specific processes of your organization for change implementation:

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

C9. The specific processes of your organization for *Configuration Status Accounting* (i)*:*

Configuration Status Accounting is a function for managing the capture, recording and reporting of information needed to manage configuration items effectively. Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

C10. The specific processes of your organization for *physical and functional audits* (i) in specific time intervals:

Physical Configuration Audit is the formal comparison of the actual (as-built) configuration of an item with the related product definition information. The PCA verifies the configuration item meets the documentation and establishes the product baseline.

Functional Audit is a formal examination to verify that a configuration item has achieved the functional and performance characteristics specified in its functional and allocated configuration documentation. Verifies the product configuration fulfills the functional requirements Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

C11. The allocation of process owners to CM processes in your organization

Process Owner: A person resposible for stakeholder guidance and feedback management as well as process monitoring and maintenance with respect to processes Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

C12. The easy access of stakeholders in your organization to latest CM process documentation:

- Initial ☐
- Managed ☐
- Standard ☐
- Optimizing ☐

C13. The possibility to customize the CM processes in your organization for various project types:

- Initial ☐
- Managed ☐
- Standard ☐
- Optimizing ☐

Section D: Information Technology

D1. Is Configuration Management supported with an information system in your organization?

- Yes, a standard IT tool for the whole organization ☐
- Yes, different IT tools used by different groups/projects ☐
- No ☐

D2. What IT tool(s) is/are being used in your organization to cover CM functionalities:

D3. **Initial, Only some individuals/groups might autonomously consider a few aspects**

Managed, no generic formal approach in place but some aspects might be applied in some projects

Standard, generic formal approach in place and most aspects are applied to most/all projects or operations activities

Optimizing,all aspects are applied to the entire organization and continuously being updated and improving

In the spirit the four possible levels of maturity shown above, how would you characterize?

--

The integration of the CM IT tool(s) with the other IT systems being used in the organization:

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

D4. **Consideration of user-friendliness of the CM IT tool(s) when they have been chosen in your organization :**

- Initial ☐
- Managed ☐
- Standard ☐
- Optimizing ☐

D5. **The entire change management capability of the IT system and its utilization in your organization? (e.g. change impact analysis, ECP and ECO creation, etc):**

- Not capable ☐
- Initial ☐
- Managed ☐
- Standard ☐
- Optimizing ☐

D6. **The capability in the IT system for tracking the change status and its utilization in your organization:**

- Not capable ☐
- Initial ☐
- Managed ☐
- Standard ☐
- Optimizing ☐

D7. **The capability of the IT system for tracing changes to requirements, products and responsible personnel and its utilization level in your organization:**

- Not capable ☐
- Initial ☐
- Managed ☐
- Standard ☐
- Optimizing ☐

D8. **The capability of the IT systems for Identification, Naming and Versioning and Variant Control and its utilization in your organization:**

- Not capable ☐
- Initial ☐
- Managed ☐
- Standard ☐
- Optimizing ☐

D9. **The capability of the IT system for accessing and managing workflows and its utilization in your organization:**

- Not capable ☐
- Initial ☐
- Managed ☐
- Standard ☐
- Optimizing ☐

D10. The capability of the IT system for authorization of accessing and modifying CM-related documents (i) and its utilization in your organization:

Documents such as specifications, design parameters, process documents, reports, etc

- Not capable ☐
- Initial ☐
- Managed ☐
- Standard ☐
- Optimizing ☐

Section E: Organization and Value Stream

E1. To which organizational unit is your CM activities allocated e.g. Quality Management, Engineering, standalone CM unit, etc?

E2. **Initial, Only some individuals/groups might autonomously consider a few aspects**

Managed, no generic formal approach in place but some aspects might be applied in some projects

Standard, generic formal approach in place and most aspects are applied to most/all projects or operations activities

Optimizing,all aspects are applied to the entire organization and continuously being updated and improving

In the spirit the four possible levels of maturity shown above, how would you characterize?

--

The consideration of organization complexity and needs in the choice of CM organization structure in your organization:

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

E3. **The description of concrete roles and responsibilities for CM-related activities in your organization:**

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

E4. The promotion of cross-functional collaboration in CM-related activities among stakeholders:

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

E5. **The consideration of CM tasks to be performed by subcontractors during contract negotiations:**

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

E6. **The involvement of key stakeholders (e.g. customers) in major requirement changes:**

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

Section F: Knowledge and Support

Initial, Only some individuals/groups might autonomously consider a few aspects

Managed, no generic formal approach in place but some aspects might be applied in some projects

Standard, generic formal approach in place and most aspects are applied to most/all projects or operations activities

Optimizing,all aspects are applied to the entire organization and continuously being updated and improving

In the spirit the four possible levels of maturity shown above, how would you characterize?

F1. The availability of a standard set of CM terminology and knowledge support which is accessible by all CM stakeholders:

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

F2. The availability of regular training activities in the area of CM (e.g. process, IT and content trainings):

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

F3. The accessibility and usage of the CM-related standards, lessons learned and best practice scenarios:

Initial ☐

Managed ☐

Standard ☐

Optimizing ☐

F4. **The support and empowerment for CM disciplines from the top management:**

- [ ] Initial
- [ ] Managed
- [ ] Standard
- [ ] Optimizing

F5. **The communication of obtained benefits from CM activities by top management:**

- [ ] Initial
- [ ] Managed
- [ ] Standard
- [ ] Optimizing

F6. **The utilization of internal and external benchmarking in different CM dimensions:**

- [ ] Initial
- [ ] Managed
- [ ] Standard
- [ ] Optimizing

Section G: Demographic

G1. **In your opinion how much did this questionnaire cover different aspects of Configuration Management? Please specify if you believe something was missing.**

G2. How much in your opinion do Configuration Management activities affect the leadtime and product cost during the product lifecycle?

CM has direct positive impact on lifecycle costs and leadtime ☐

CM has indirect positive impact on lifecycle costs and leadtime ☐

CM has negative impact on lifecycle costs and leadtime ☐

CM has no impact on lifecycle costs and leadtime ☐

G3. Considering your experience how much do you believe Configuration Management can affect *Reliability, Availability, Maintainability and Safety* (i) of the product/Facility?

Reliability: The ability of a system to perform its required functions under stated conditions for a specified period of time - Shown by MTBF (Mean Time Between Failure)

Availability: The proportion of time that a system is in a functioning condition - Shown by (Mean Uptime)

Maintainability: The ease with which maintenance of a system can be performed in accordance with prescribed requirements - shown by MTTR (Mean Time To Repair) Safety: A state of a system in which the risk of impairment as well as hazards caused by these risks are not existing or are controllable

Has a considerable positive impact on all factors ☐

Has a limited positive impact on all factors ☐

Has negative impact on all factors ☐

Different impacts on different factors (please specify) ☐

Different impacts on different factors (please specify)

G4. Please specify how relevant is CM to your current job and at what position you are currently working in your organization?

- Directly involved in CM activities ☐
- Indirectly involved in CM activities ☐
- Not involved at all ☐

G5. Please specify the type of organziation you are representing?

- Public sector ☐
- Private sector ☐

G6. Please specify the size of the organization you are representing?

- 10 employees or less ☐
- Between 10 to 50 employees ☐
- Between 50 to 250 employees ☐
- Between 250 to 1000 employees ☐
- more than 1000 employees ☐

G7. Please specify which industry segment you organization belongs to?

Automotive ☐

Aerospace ☐

Healthcare ☐

Food industry ☐

Consulting services ☐

Education ☐

Energy ☐

Communication ☐

Clothing ☐

Other ☐

Other

G8. **If you would like to have an individual and private comparison of your organization with the related industry sector in addition to the report, please provide your organization name.**

Thank you very much for your time

Appendix B. Engineering Change Management literature review

B.1 Engineering Change Management Overview

B.1.1 Definition of Engineering Change

Within the literature, several different defintions of EC are given, which highlight slightly different aspects (Jarratt, et al., 2010). It is therefore necessary to establish a common understanding of the term and to identify the key aspects. These are important for a comprehensive description of Engineering Change as a process within an organization.

One of the definitions proposed is that of Wright (Wright, 1997), which states that „an engineering change (EC) is a modification to a component of a product, after that product has entered production“. While this definition makes a distinction between what is traditionally understood as product development and the EC activity, the division appears to be artificial. A further shortcoming of Wright's approach is the fact that all alterations occuring during the design and development phase of a product are neglected and out of the scope of EC and the processes and tools supporting the further development of existing versions of a product (Jarratt, et al., 2004).

A further definition is offered by Huang and Mak (Huang & Mak, 1998), who describe EC as "the changes and modifications in forms, fits, materials, dimensions, functions, etc. of a product or a component." While Huang and Mak do define the scope of the change, they fail to mention the time at which the change can occur in order to fall into the category of EC (Jarratt, et al., 2010).

Terwiesch and Loch offer the definition „engineering change orders (ECOs)-changes to parts, drawings or software that have already been released“ (Terwiesch & Loch, 1999). An important aspect this approach adds is its inclusion of software, which has by now become a major part of many products. Furthermore, it makes a clear distinction between a change process and many other forms of iteration, by specifically requiring a product be already released and that a task that has already been completed be revisited in order to undergo an EC (Wynn, et al., 2007) (Jarratt, et al., 2010).

It is important to mention that none of the references mention the origin, size or scope of the change and thus can include such diverse endeavours as „a small revision of a diagram taking one engineer a few minutes to a major redesign operation involving a large team of engineers working over a period of many months or even years“ (Jarratt, et al., 2010). Thus, Jarratt et al. (Jarratt, et al., 2004) have provided a more complete definition of EC based on Terwiesch and Loch (Terwiesch & Loch, 1999) which is going to be adopted for the purpose of this paper. „An engineering change is an alteration made to parts, drawings or software that have already been released during the product design process. The change can be of any size or type; the change can involve any number of people and take any length of time.“ (Jarratt, et al., 2004).

B.2 Distinction of Engineering Change

Having proposed a clear definition, it is important to make a clear distinction between EC and other terms often used within organizations. Most obvious is the need to differentiate the term Engineering Change Management from the frequently used Change Management. The term Change Management is widely accepted to describe the administration and supervision of transformation processes within an organization. It involves a structured approach of moving from a current state to a future state by implementing measures as diverse as minor changes in existing processes, up to pervasive changes, which can even cause deep amendments to an organization's strategy (Strazdina & Kirikova, 2011). Furthermore, a large body of research exists on change processes involving organizational mergers and acquisitions. Within this context, the main concern is an integration and mediation process, mainly regarding the employee culture and acceptance. This is achieved through a combination of planning, management, personnel planning, control and leadership (Körfer, 2006). Being aware of the tremendous differences between the concept of Change Management, as referred to within many organizational contexts, one can clearly see the need for making a clear distinction to the Engineering Change Management process, which is concerned with the alteration to products, not organizations and business processes.

Another term, which is often heard in connection with ECM is that of Configuration Management (CM). Having been developed in the 1950s by the US military in order to control the documentation in the manufacturing of missiles (Gonzalez, 2002), Configuration Management is an approach that is widely used in safety critical systems, such as aerospace and nuclear projects. The main driver behind Configuration Management is to address problems occuring in projects due to unchecked changes in one of the sub-systems which results in consequences in the wider context of the system. Thus, the aim is to analyze the consequences of any change before it is made and to provide traceability of product data in order to understand causes of problems, as well as to diagnose and resolve any problem arising through the adjustment of any part of hardware or software. The main focus of CM lies in the processes, procedures and users keeping the data intact throughout the lifecycle by controlling and recording the changes in a data system (Lindkvist, et al., 2013). The current definition in the IEEE standard (Institute of Electrical and Electronics Engineers, 1990) describes CM as "a discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements". As can be seen by comparison of this definition with the previously stated characterization of ECM, the Engineering Change process is a sub-category of the Configuration Management process.

B.2.1 Engineering Change within Product Lifecycle Management

One of the most common images within economic theory is that of the Product Lifecycle consisting of different phases from product introduction to phase out.

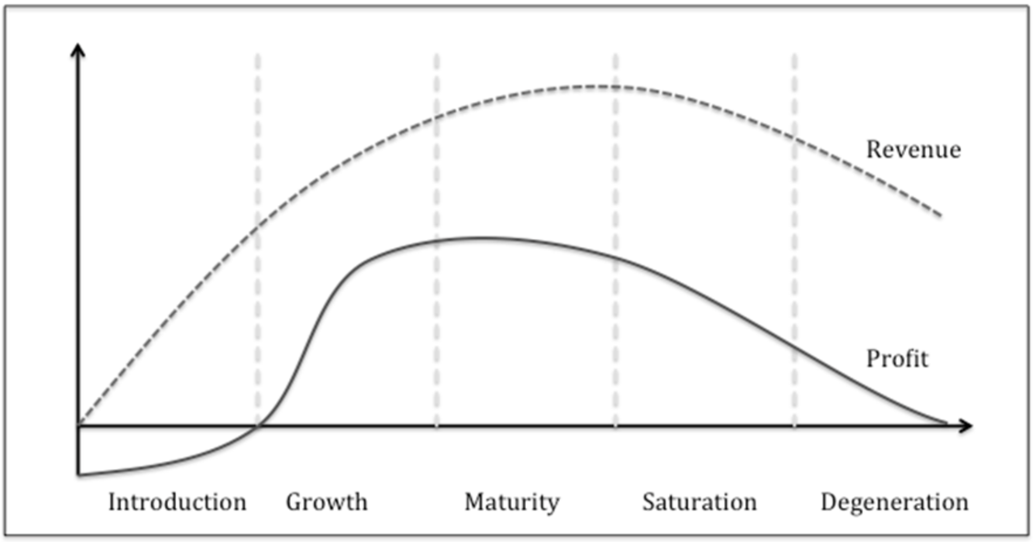

Figure 52 – Revenue and Profit over the Course of the Product Lifecycle

While it is essential for any company to follow economic imperatives, further facets of a Product Lifecycle have been gaining prominence in theory and practice. Most prominent is the introduction of the “Triple Bottom Line” approach, which does, apart from the economic evaluation, also include factors of social and environmental sustainability (Elkington, 1997).

Tripple Bottom Line	Description
Social	Skills, motivation and loyalty of employees and business partners. Vaule is added to the community in which a company operates.
Environmental	Reduction of the consumption of natural resources below the natural replenishment rate.
Economic	Guaranteed cash-flow at any time while producing return for the shareholders.

Table 6 – Tripple Bottom Line – Based on (Elkington, 1997) and (Gmelin & Seuring, 2014)

Producing a return for the shareholders while guaranteeing a cash-flow is one of the essential goals of a company in order to ensure its economic viability. To reach environmental sustainability, the consumption of natural resources must be reduced below the level of natural replenishment. The goal of social sustainability is not as clear cut and includes various factors, such as the skills and motivation of the employees, loyalty to the company and added value to the community (Gmelin & Seuring, 2014).

One can also view the product lifecycle from an IT standpoint. There are several phases, at which information regarding a product are produced. The data is to be integrated and made available to decision makers. The first phase is the product planning, which specifies the field of the product development. Following is the actual development effort, which entails a full description of the product. The production planning is the next step, which is then followed by the actual production. Sales, product use and service, as well as recycling, all produce different kinds of data and require for their completion the integration of information from previous phases of the lifecycle.

As has been described, organizations aim at reaching not only economic viability, but also intend to reach social and environmental sustainability. Furthermore, companies require the right information concerning the different steps of the lifecycle in order to maintain competitiveness. In managing the goals an organization has set for the life cycle of a product, the task of Product Lifecycle Management (PLM) is necessary. PLM is an approach for the integration of information on activities, actors and attributes of products in various phases of the product life cycle. Included in this are the adequate tools, standards and technologies for product development throughout the entire lifecycle. The key to effective PLM are the individual stakeholders and the communication and relations between them (Denger & Unzeitig, 2012). In order to achieve this, PLM systems must offer stakeholders adequate information to support their decisions without significantly increasing the effort for administration (Softic, et al., 2014).

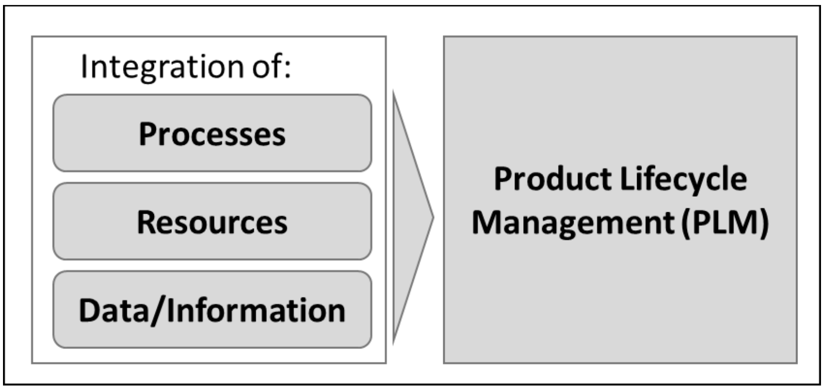

Figure 53 – Product Lifecycle Management

Within the context of Product Lifecycle Management, the Engineering Change process is essential for managing any adjustments, which arise after a part, drawing or piece of software has already been released. ECM is therefore a sub-process of PLM, which is necessary in order to achieve a continous business process with optimized product lifecycles. In order to do so, many companies have adopted a standardized process, which will be described in the following section.

B.3 The Engineering Change Process

It is the goal of the subsequent section to further enhance the understanding of the Engineering Change process and some of the key factors determining the nature of a change initiative. Also, the impact a change can have on various parts of an organization is explored in more detail.

B.3.1 Model of an EC process

In existing software applications for PLM such as Teamcenter, a very general approach to ECM is already provided. In accordance to the CMII standard (Guess, 2013), a comprehensive process for the initiation, administration, review and approval, as well as for the execution of product changes on an enterprise basis is already implemented (Siemens Product Lifecycle Management Software Inc., 2010). Within the bounds of such a software tool, any process model can be realized, as to accomodate the varying demands of organizations in different fields (Guess, 2005). Some guidelines for a Configuration and Quality Management approach include the ISO10007 (International Organisation for Standardisation, 2003), as well as the ISO9001 (International Organisation for Standardisation, 2008).

Even though the EC processes of individual organizations appear similar on a high level, a detailed analysis yields significant differences, depending on the requirements (Pikosz & Malmqvist, 1998). It is thus not surprising to find a large number of proposals for a generic EC process described in the literature. Within the context of this work, two of the generic models are described in detail. For a more comprehensive review, one can refer to Jarratt et. al. (Jarratt, et al., 2010).

Dividing the EC process into six distinct steps, Jarratt et al. propose a complete approach, which will – with minor additions – serve as the basis for this thesis (Jarratt, et al., 2004).

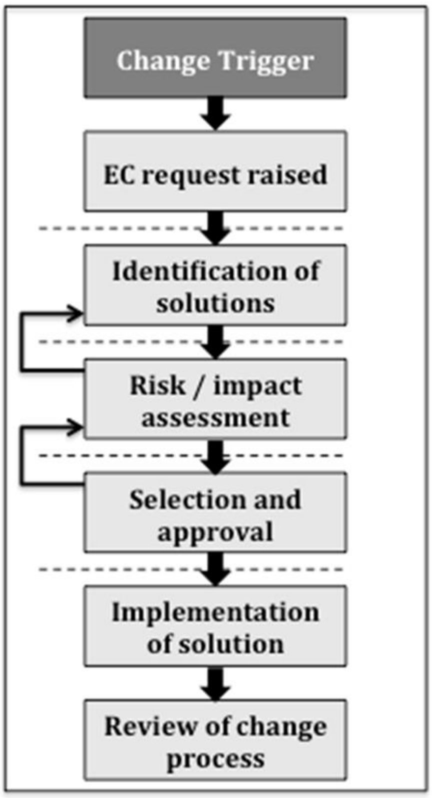

Figure 54 – Process Model of the Engineering Change Process – Based on (Jarratt, et al., 2004)

The process entails the steps as set forth in Figure 54:

1. As basis for the initiation of the EC process, a request must be made. It is commonplace to have standard forms to be completed, which can be paper-based or in electronic form as part of a software system. During this stage, the person must clarify the attributes of the change, such as the part or component pertinent to the change, the priority, the type of change, etc. This information, hereafter referred to as an Engineering Change Request (ECR), is then transferred to the next stage.
2. The idenfification of potential solutions to the request for change is the subsequent step. Many times, only one solution is examined, due to restrictions in time, obvious solutions or the fact that engineers stop investigating once a viable solution is found. The results of this porcess are the Engineering Change Proposals (ECPs) and are, upon completion, forwarded to the next entity.
3. In assessing the risk of implementing a solution, many factors must be considered. It is thus imperative to incorporate assessments of the impact on design and production schedules, relationships with suppliers and budget. It is important to mention at this point that the further along a development process a change is implemented, the higher the effort and cost for an adjustment will be.
4. Having selected a potential solution, a formal approval process is initiated. In many organizations, this is carried out in an Engineering Change Board (ECB), which is ideally made up of senior staff from all the key fuctions related to the product, such as marketing, supply chain, product design,

manufacturing, quality assurance and product support.

5. Upon acceptance, an Engineering Change Order (ECO) is issued and the implementation is scheduled. Depending on the priority of the change, an immediate realization can be required. In many cases though, the change will be phased in. While the ECO can be regarded as a document which describes an approved engineering change to a product and is the authority or directive to implement the change into the product and its documentation" (Monahan, 1995), it is also imperative to ensure the proper documentation and distribution of the change via an Engineering Change Notice (ECN), which is distributed throughout the organization.
6. The final step of the process is the review and evaluation of the initiated change after a period of time. In order to evaluate and improve the change process, not only aspects concerning the function of the product should be considered, but also a complete picture of the impacts of the change should be made. Even though not all companies do carry out this form of formalized review process, it is essential within the context of this work.

It is important to note that at certain stages within the process, iterations may occur. Thus, if the evaluation at Phase 3 of the risks and impacts of a solution are not acceptable for an organizaiton, the process will return to Phase 2 and other possible solutions are then to be identified. Similarly, if at the approval phase, the ECB decides that further analysis is required in order to make a sound decision, the process will return to the risk and impact analysis phase. Although the described two scenarios cover most of the relevant situations, one can of course imagine further possible iterative loops, which are not represented here. Of these, the most extreme case is that during the review of a particular change process, it is discovered that the implemented solution has not solved the original issue or made things worse. In this case, the process would return to its original stage and a new change request would have to be raised.

As can be seen in Figure 54, the engineering change process features four breaking points, at which the process can be brought to stop. In such cases, no reiteration occurs, but rather a conscious decision on giving up this particular initiative is made.

At the stage of the evaluation of the change after a period of time, only those proposals which provide an overall benefit to the company will be examined for lessons learned. Yet, it is important to note that this must include more than a mere financial analysis, as the reasons and triggers for change initiatives may be of various natures (Jarratt, et al., 2004) (Jarratt, et al., 2010).

B.3.2 Reasons for Triggering the EC process

Within the literature, many authors have proposed different views on reasons for changes to occur, such as (Pikosz & Malmqvist, 1998) and (Fricke, et al., 2000). One of the approaches, suggested by Pikosz and

Malmqvist (Pikosz & Malmqvist, 1998), divides reasons for change by the focus on either parts and documents or the stages in which the change reason arises. This is in line with the draft standard ISO 11442-6 (International Organisation for Standardization, 2006):

Focus on Parts and Documents:	Focus on Stages:
• Change of a part depending on altered function or production requirements • Change in the application of a part • Introduction of a new part • Replacement of a part • Withdrawal of a part • Corrections of errors on a document • Bringing an old document up to date	• Changes in customer specification • Faults in the interpretation of customer demands into technical requirements • Difficulties in parts fabrication of assembly • Weaknesses in the product identified during prototype testing • Quality problem with some subsystem or component • Development for future product revisions

Table 7 – EC reasons according to the draft standard ISO 11442-6 (1996)

As the preceeding categorization is utilized in the following discussion, we will adhere to the approach of Eckert et. al., dividing reasons for triggering an EC process as either emergent (coming from the product itself) or as initiated (coming from outside the product) (Eckert, et al., 2004).

Emergent changes

Emergent changes arise from the properties of the individual products themselves and include:

- Error correction: At any time during the development lifecycle, mistakes made during the design phase can be identified. Many times, these problems have to do with the product integration and can range from minor drawing errors to issues that fundamentally impair the operation of the product.
- Safety: In case products do not meet the safety requirements or are shown to be a danger to people's property, health or even life, changes are to be made. In evaluating safety issues, the producers must also keep in mind the unintended use of their products.
- Change of function: If the design does not fulfill the functional requirements set during the design phase, such as an insufficient adjustment to operating conditions, alterations are in order.
- Product quality problems: Often times, these problems can be traced back to poor design or incorrect instructions from manufacturing and assembly.

Initiated changes

Initiated changes can take the form of improvements, enhancements or adaptations to a given product. In this category, rather than listing the various possible reasons, as done in (Jarratt, et al., 2010), a focus on the directions from which an impulse to change can arise is more appropriate.

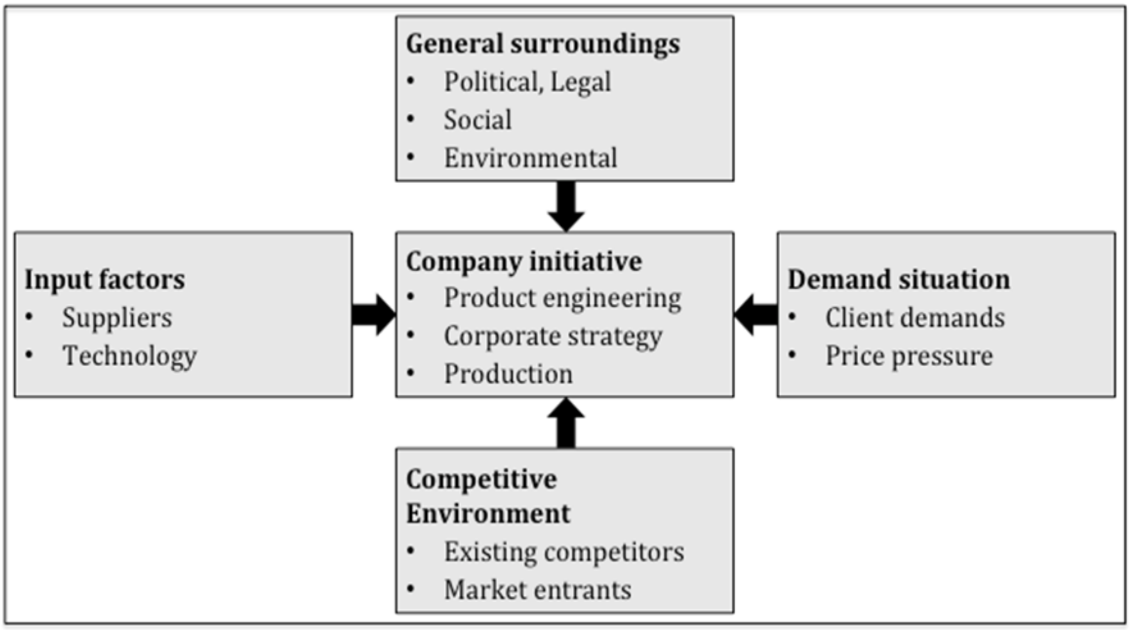

Figure 55 – Triggers of the Engineering Change Process

Products are becoming ever more complex and require a large number of input factors for their completion. In an increasingly connected supply chain, a company may even receive the impulse for change from one of its external suppliers. As outsourcing is becoming more commonplace, the changes in any sourced component also propagates into the main product. These necessary adjustments may come from problems and changes the supplier faces, as well as from technological advancements made in a field relevant to the industry.

Another area in which changes can arise, is the general conditions under which an organization operates. The most common factors are political, legal, social and environmental. Often times changes in these areas lead to altered regulations companies must to adhere to.

Further factors leading to an increased pressure to change may come from the competitive environment of a firm. In this context, it is not only important to pay attention to existing competitors, but to also keep in sight any new market entrants, which could disrupt a market and create entirely new issues to the market.

The fourth area in which reasons to change may emerge is the demand situation. Most common in this area are increased demands of customers regarding the range and magniture of performance after they have experienced the product in an application setting. Further pressures often come from a demand for lower price products, which then in turn initiates a change process.

Finally, a company itself often initiates change for internal reasons. It is thus common that product designers and engineers identify ways in which a product can be enhanced to better suit the application requirements. One of the key issues within a production firm is the manufacturability and is sometimes not sufficiently regarded in the design process. This situation may later be rectified by initiating a change process. A further source of initiated change is the company strategy, which can have large impacts on make-or-buy decisions, which greatly influence the overall design of any product.

B.4 Factors Impacting the EC process

There are numerous factors influencing the efficiency and effectiveness of an engineering change process. The following section will provide an overview of the determining factors and provide a framework for classification of Engineering Change cases.

B.4.1 Classification of EC based on Urgency and Timing

One can not only distinguish change processes by whether they are emergent or initiated, as done in the previous section, but can also derive valuable insights by putting an initiative in perspective concerning timing and urgency.

In characterizing an engineering change initiative by urgency, one can better understand the prioritization it is given in the overall context of an organizations engineering activities. One of the commonly used approaches is that of DiPrima (DiPrima, 1982), which categorizes levels of urgency as follows:

- Immediate: Some engineering changes are to be implemented immediately, as they are safety critical or are essential for the continued goodwill of customers.
- Mandatory: While some flexibility on the timing is granted, these changes should be undertaken as soon as is feasible.
- Convenience: When implementing this typy of change, one should be mindful not to disturb the ongoing production, as these changes only result in incremental improvements to the product and can be implemented whenever practical.

Within the literature of Product Lifecycle Management, it is widely recognized that the earlier a design flaw is discovered, the smaller the impact on the production and the easier a change can be implemented.

Therefore, a strategy called „frontloading“ is becoming more popular, whereby organizations increase the engineering ressources early in a project, in order to shorten development time and decrease the impact of adjustments on the subsequent phases of a project (Maropoulos & Ceglarek, 2010).

Especially for long-lead-time products, Reidelbach (Reidelbach, 1991) suggests a categorization of engineering changes depending on the timing within the product development process:

- Early, low impact: As the issue is addressed before the design freeze and before any procurement is officially undergone, the impact is generally low.
- Mid-production: As the EC occurs after start of manufacturing and the approval of procurement activities, there could be a tremendous impact on long-lead-time projects.
- Late, expedited EC: This situation is potentially very damaging to the completion of a project, depending on which of the sub-systems is affected.

B.4.2 Engineering Change and Product Characteristics

One of the key factors for how a change process unfolds is the nature of the product. The three most influential aspects are the complexity of the product, the architecture of the product and the degree of innovation within a solution (Jarratt, et al., 2010).

Product complexity:

There are many ways to conceptualize product complexity. For example, one can judge the complexity of a system by the amount of information inherent within it (Suh, 1990). Yet, the key type of complexity when it comes to an engineering change perspective is that of connectivity, which describes how elements are connected and interact within the entire system (Sosa, et al., 2007). In products with many dependencies and interactions among the elements, changes in one part are very likely to propagate additional changes. As a result, the change process becomes increasingly hard to control as the number of parameters and their impacts on each other increase (Fricke, et al., 2000).

There exist various ways to describe a product by the way it is broken down. Most commonplace is the decomposition of any product into components and parts, also described as the multilevel bill of materials (BOM). There are also other ways of breaking down a product into other BOMs. Yet, none of these approaches fully cover the entirety of interconnections between the individual components. Nonetheless, when implementing a change, one needs to consider the whole spectrum of information available. This is where some of the PLM solutions fall short, as they provide information tailored to the needs of particular users.

Especially in the case of complex products, the manufacturer must decide on how to tackle the issue of inevitable engineering change. According to Barber et al. (Barber, et al., 19), three options present themselves:

1. Design to meet current requirements
2. Design to meet the predicted requirements at the end of a product's lifespan
3. Design so that the product can be updated to meet future needs

The first option leaves little room for easy adjustments and could result in very costly modifications to meet future requirements. Designing a product to meet the predicted demands of the entire lifecycle could result in over-engineering and thus runs the risk of adding, at great cost, features that are not in demand. Therefore, the alternative of designing a product with future updates in mind, which leaves one with an increasingly modular product architecture, appears to be the most universally applicable approach (Jarratt, et al., 2010).

Product Architecture:

The fashion in which a change impacts a product is fundamentally linked to how it is built. This is the product architecture which is defined as (1) the arrangement of functional elements; (2) the mapping from functional elements to physical components; (3) the specification of interfaces among the interacysical components" (Ulrich, 1995).

One can distinguish two archetypical configurations of product architecture, namely modular and integrated. In practice, almost all products are situated somewhere in the spectrum between the two extremes. In recent times, the trend in many industries has been to create adaptable and competitive products by promoting modularity. The modular design has the advantage of being easily adaptable to changing requirements, as long as the interfaces remain the same. However, in case it becomes necessary to change the interfaces, a dramatic increase in the magnitude of the change is to be expected (Jarratt, et al., 2010).

In relation to the propagation of change, one can make another categorization of products into absorbers, carriers and multipliers (Eckert, et al., 2004):

- Absorbers: These can be further distinguished into being either „partial" or „total". Where a total absorber causes no further change, while accomodating a number of changes, it is more realistic to have a partial absorber that may contain many changes and pass on only a few.
- Carriers: The change problem is neither reduced nor increased. A mere transfer of the change from one component to the next takes place.

- Multipliers; The situation is made more complex, as the modification of one component may cause an „avalanche“ of further changes.

In some cases, a component may be an absorber up to a certain extent, at which point its tolerance to accomodate change is exceeded. At this point, this very component may change into a multiplier, as the change of the component outside the aforementioned limits entails the necessity to modify many other parts of the product (Jarratt, et al., 2010).

Degree of innovation:

Highly innovative products are much more likely to require changes. This is due to the fact that in the beginning of a product's life-cycle, a low degree of data and knowledge is available on such products.

Among the key factors in this context is how an organization runs its research and development and how the portfolios are managed. It has been suggested to decouple the innovation process from the product development and to test new technologies in a prototype before starting to develop a market-level product (Jarratt, et al., 2010). Nonetheless, Eckert et al. point out that there could still be problems when multiple new technologies are introduced into the product design process (Eckert, et al., 2009).

B.4.3 Organizational Impact Factors

The success of a change initiative is not only a matter of employing the proper technique and technology. The involvement of the right people and the general attitude towards change are just as essential as the backing of an initiative by the management.

Company structure and culture:

Among the key factors for successful ECM is good communication between the stakeholders. Clark and Fujimoto indicate that up to two-thirds of all technical changes could be prevented by better communication and discipline (Clark & Fujimoto, 1991). Failures in communication can result in a large number of changes and high cost changes as data, which is not up-to-date, is being used in the decision making process.

In addition, a successful change initiative requires the backing of the proper stakeholders within management in order to obtain the resources necessary to implement the introduced change. It is thus necessary to be mindful of the support required, while at the same time maintaining a level of control on initiatives through a party outside the engineering chain of command.

Another key issue is decision discipline, which has been focused upon as a cause of changes and a reason for change propagation. Poor decision discipline occurs if decisions are made without basis or when necessary decisions are postponed. This issue is closely linked to the company culture, and reasons for postponed decisions are (Fricke, et al., 2000):

- Initiating change is seen as unpleasant, as it draws unwanted attention to the initiator and his department.
- People wait for other changes to occur in order to shift the responsibility.
- Individuals want to be seen as heroes by rescuing a project which is on the verge of failing.

Attitudes towards engineering change:

Generally, engineering change processes are regarded as being disruptive, as alterations disturb the smooth running and timely delivery of a product. In part, this is due to the fact that manufacturing processes must be adjusted and down time is to be expected.

Interestingly, Eckert et al. found that, while engineers mostly resent changes arising from mistakes, change initiatives leading to an enhancement or the adaptation of a product, were quite welcomed, even if very similar tasks were involved in realizing the change (Eckert, et al., 2004).

B.4.4 Effects and Impacts of Engineering Change

The previous sections have addressed various factors influencing the nature of the change process. Yet, for EC decision making, the effects and impacts of making a change are of the upmost importance.

As pointed out, the timing of the change initiative can have a tremendous influence on the planning, scheduling and cost of a project (see B.4.1). In addition to this, personnel and organizational issues set the organizational framework in which a change is to occur and are essential for the successful management of change in an organization (see B.4.3). One of the most essential factors defining the nature of the change are the product characteristics (see B.4.2).

In general, making a change to a product affects planning, scheduling and project cost and can jeopardise the success of projects. Especially important, as already stated, is the fact that changes can propagate. This is the case when a change in one component spreads to other parts of the product. It is important to note that these issues can then be even amplified by difficulties in the supply chain, as time delays in the delivery of specifications or components compresses an already tight production schedule (Jarratt, et al., 2010). Terwiesch and Loch have identified three couplings that are likely to lead to propagation (Terwiesch & Loch, 1999):

- Propagation from components to manufacturing
- Propagation between components within the same sub-system

- Propagation between components in different subsystems

In a more general approach to the effects of a change initiative, Riviere et al. propose the following aspects to be considered (Riviere, et al., 2002):

- Near Term Cost Impacts: Being able to estimate the cost of a change initiative is essential to good decision making.
- Impacts on Schedule: The adjustment of activities and processes does many times delay the the originally planned proceedings and requires a re-scheduling.
- Impacts on Product Performance: One of the key drivers for change is the enhancement of product performance. Nonetheless, some changes, which are especially targeted at enhanced manufacturability or cost can have adverse consequences on product performance.
- Impacts to other Components: The changes in one component can propagate through the interfaces and trigger further change.
- Impacts on Supply Chain: As the quality or quantity of a purchased part change, adjustments in the supply chain may become necessary.

Considering the complex interdependencies that become evident when initiating a change in one area, many organizations have turned to special processes and IT tools for support. These will be subsequently introduced.

B.5 Tools Supporting the EC Process

Tools and methods supporting the engineering change process can roughly be divided into two categories. Firstly, there are the tools that support in the management of the workflow and documentation of the process. The other category of support tools aids engineers at making decisions at a certain point within the change process (Jarratt, et al., 2010).

B.5.1 Work Flow and Documentation Support

In the past, some firms have relied on a paper-based ECM system, which only allowed one person or group to access the data and document at a time. With the increased volume of changes arising for a product and multiple changes being carried out simultaneously, such systems have become inefficient (Kidd & Thompson, 2000). Thus, firms are now utilizing computer-based systems generating change requests and change notice forms, which are then processed further electronically.

Apart from decision support systems, which will be discussed in a following section, Huang and Mak distinguish between three categories of computer based tools (Huang & Mak, 1998):

1. Dedicated Engineering Change Management systems contain the databases of engineering change activities and generate the electronic workflow documents, such as the change request, the change proposal, as well as the engineering change order. Such systems were often developed in-house and have a very limited flexibility.
2. Computer aided Configuration Management (CM) systems incorporate the functionality of dedicated systems, but additionally allow for product structuring and versioning.
3. Product Data Management (PDM) or Product Life-cycle Management (PLM) systems have become increasingly commonplace within organizations and are used throughout the entire design and product life-cycle. In addition to Configuration Management functionality, these systems support all stages of the product life-cycle. There are a number of large software companies providing such tools, including Siemens with its „Teamcenter“ suite, Dassault Systems with „Enovia“, Oracle with „Oracle Agile PLM“, Parametric Technology Corporation with „Windchill“ and SAP with „SAP PLM“. It is at this point important to note that, due to the complexity of such systems they are all developed by large corporations. Nonetheless, in the field of ECM, the „processes that have been implemented through workflow in PDM systems seem to be a copy of the old paper-based systems“ (Riviere, et al., 2002).

The many existing capabilities of PLM systems are a promising indicator that, once a feasible approach is developed, decision support can be integrated and greatly improve the handling of engineering changes within organizations. This is due to the fact that, only within the context of the entire life-cycle, one can get an overview on time, stakeholders, components and resources used (Softic, et al., 2014). The deployment of enterprise-wide management and executive information systems can thus ensure organizational agility and competitiveness (Noran, 2009).

B.5.2 Soft Technologies for Decision Support

One of the most effective ways of avoiding costly changes within the life-cycle is to engage engineers into discussions about the key aspects of the products, such as customer requirements, manufacturability and potential failure modes, as early as possible. In doing so, many firms rely upon one of the following soft measures (Huang & Mak, kein Datum):

- Design for Manufacture and Assembly (DFMA) is used to help design teams fulfill the requirements on a product with the simplest means of product design and production. By employing this approach, firms aim at identifying and eliminating the over-engineering in product and production design.
- Failure Mode and Effects Analysis (FMEA) is a measure in preventative quality management and has the goal of identifying potential defects in a product as early in the design process as possible.

- Quality Function Deployment (QFD) is a quality assurance method representing a framework for implementing quality requirements at every stage of a product design process. This approach aims at just fulfilling customer requirements without over-engineering.

The mentioned methodologies are helpful in guiding a design team toward increased engineering efficiency and effectiveness. Nonetheless, extensive expertise is required for the proper application and many aspects within the engineering process are not considered. That is why numerous organizations have adapted Decision Support tools to assist in the Engineering Change process.

B.5.3 Decision Support Tools

Utilizing software tools to support the decision-making process within engineering is very common for companies in the industrial sector. Yet, there are great variations in functionality and the context in which they operate. Most commonly used are Computer Aided Design (CAD) packages which can assist designers in the recognition of errors. It is important to point out though, that these tools, including Siemens NX and CATIA, are only able to detect the first stage of change and are unable to predict the propagation of a change to further components.

In trying to fill the gap of decision support in engineering change, many academic tools and prototypes have been developed. Yet, none of the tools seem to have been thoroughly tested or even implemented on a large scale. The following table will introduce some of the approaches and point out the individual shortcomings, which have impaired a widespread adoption.

Tool	Approach	Shortcomings
CAD - Computer Aided Design	Analysis of geometry and indication of mismatch between design changes and geometrical constraints	Change propagation not regarded; functional relationships between components not assessed
C-FAR (Change Favourable Representation) (Cohen, et al., 2000)	Examination of attributes and interactions between the core elements of entities in product models to predict the propagation effect	Suitable only for simple or small products, as limited interactions can be represented
Redesign IT (Ollinger & Stahovich, 2001)	Generation and evaluation of proposals for redesign plans by using a product model with physical quantities and causal relationships	Only specification of direction of change; no quantitative suggestions (Ollinger & Stahovich, 2001)
CPM - Change Prediction Method (Clarkson, et al., 2001)	Utilization of Design Structure Matrix (DSM) to estimate risk of change propagation within a product	Complex adjustment for varying cases required
CECM - Collaborative Environment for Engineering Change Management (Lee, et al., 2006)	Case-based reasoning retrieval of previous change initiatives based on different ontologies; Basis for collaboration with structured online workflows to capture and integrate informal and unstructured knowledge	No adaptation of retrieved cases to match current change initiative
ADVICE (Kocar & Akgunduz, 2010)	Virtual environment for ECM; combines parametric and graphical information; integration of existing data for predicting change propagation and making change prioritizations	Only single criterion used for prioritization; very basic algorithms used for propagation analysis

Information Structure Framework (ISF) and Change Management Support Approach (CMSA) (Ahmad, et al., 2012)	Creation and capture of knowledge related to change propagation through structured process; prediction of propagation impact based on use of dynamic checklists	Previous knowledge not included; high effort for knowledge elicitation
FBS Linkage Method (Hamraz, 2013)	Integration of functional reasoning and change prediction based on a network of functional, behavioral and structural attributes	Regards only impact of change and change propagation on cost measures
Ontology based tool for task tracking and decision support (Softic, et al., 2014)	Acquisition, editing and querying of information based on ontological representation of product data; Visualization of task interdependencies	Prediction of change impact with only rudimentary tools; knowledge from past projects not sufficiently integrated

Table 8 – Tools for Engineering Change Management Decision Support

Despite the many attempts at providing a decision support tool for the engineering change process, all of the candidate approaches have their respective shortcomings. One of the results of the preceding analysis is the fact that the mentioned tools are either too general to add concrete support in the decision making process or too narrowly focused on issues like the change propagation analysis. Of course, there is a trade-off between the accuracy of a model and its complexity, but the proposed systems lack the flexibility to adjust to any specific environment. Most importantly, the candidate approaches do not sufficiently include the tacit knowledge already generated in previous cases.

Appendix C. Knowledge Management and Decision Support Systems

C.1 Knowledge and Knowledge Management

In the current business environment, companies face ever increasing complexity and dynamics of change. Therefore, firms have to adjust ever more quickly and must find ways to increase agility and decision quality. In doing so, Knowledge Management (KM) has become a domain of great interest, as it „allows the making of predictions, casual associations, or descriptive decisions about what to do“ (Bohn, 1995).

Knowledge is defined in very different ways within the literature. While it has been suggested that knowledge is „information combined with experience, context, interpretation and reflection“ (Davenport & Prusak, 1998), Harris offers a definition of knowledge more closely related to EC as „the output of product development process“ (Harris, 2009).

C.1.1 Types of Knowledge

Properly distinguishing knowledge from other concepts mentioned in this area, a hierarchical concept of data, information, knowledge and wisdom is proposed.

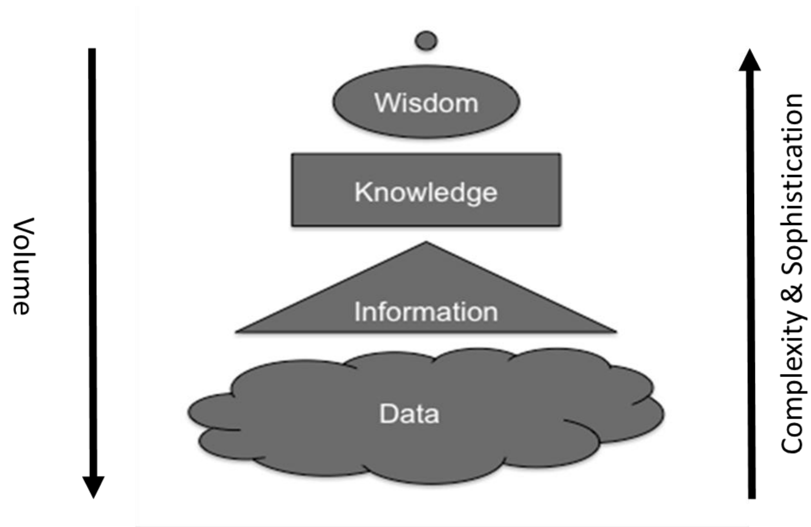

Figure 56 – Knowledge Hierarchy (Sajja & Akerkar, 2010)

Within the context of this model, data is interpreted to provide information. Data without interpretation and additional processing is of no use, which is why, in order to obtain knowledge, one must take the necessary step through information. Yet, information is a necessary medium or material for eliciting and constructing knowledge but not necessarily embodies knowledge (Harris, 2009). Through the analysis of information, one can thus derive knowledge, which can over the course of time and accumulated experiences amount to wisdom (Sajja & Akerkar, 2010).

Another way of classifying knowledge is the distinction between tacit and explicit knowledge, which is especially important in the context of decision support. Tacit knowledge	Contextspecific, highly personal information; hard to formalize and communicate; intrinsic to experts
Explicit knowledge	Can be codified and articulated in formal language and thus be transmitted; especially important when using information technology as means of transport

Table 9 – Tacit and Explicit Knowledge

Obviously, knowledge is present within organizations in various forms and is continuously produced and converted. As a consequence, an enterprise-wide overarching management and execution of information systems, expert systems and decision support systems is required for firms to improve the decision making. This knowledge-driven approach can greatly increase organizational agility and is essential for the survival of a firm (Noran, 2009). But in order to achieve this synergy, an integration of systems and tools is required. This way, not only the documents themselves are captured, but also the meta-knowledge linking the individual components is considered (Harris, 2009).

C.1.2 Knowledge Management Process

Within the literature, many approaches to KM have been developed, such as (Davenport & Prusak, 1998), (Kasvi, et al., 2003) and (Liu, et al., kein Datum). For the purpose of knowledge management, especially considering the product development environment, Harris has proposed a framework consisting of the following steps (Harris, 2009):

1. Identification of knowledge
2. Capture and retrieval of knowledge
3. Formalization and presentation of knowledge
4. Utilization of knowledge by integration in current projects

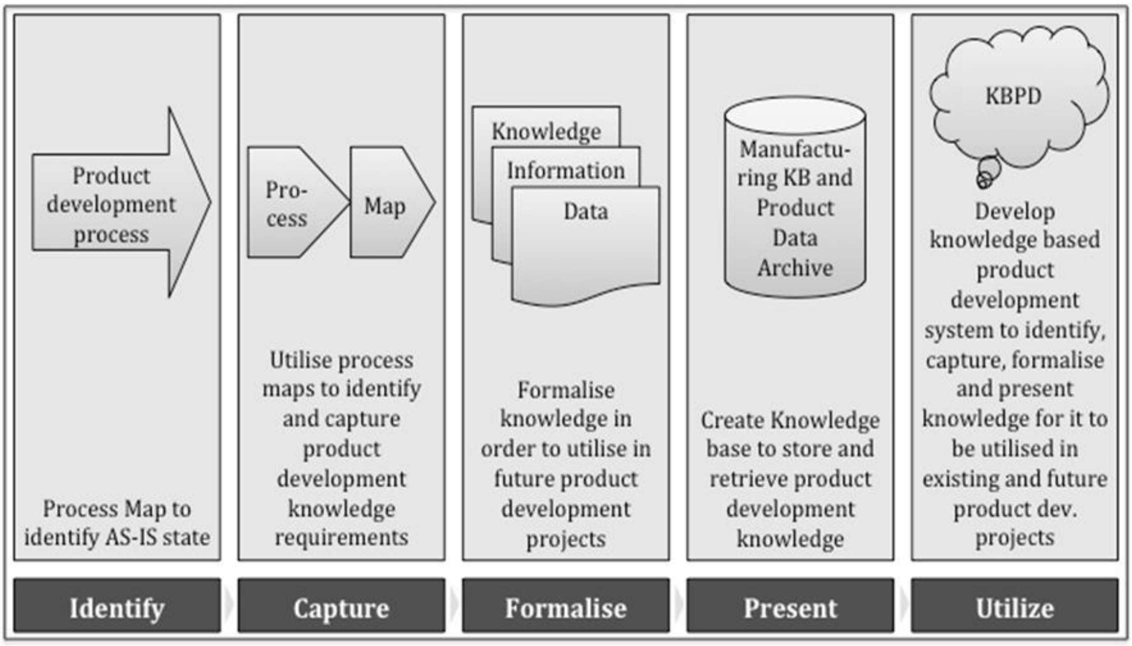

Figure 57 – Knowledge Management Process – Based on (Harris, 2009)

The proposed framework is very well suited to represent the KM process within engineering change management. As the starting point, the process map serves to identify the current state and is the input for capturing the product development knowledge. The obtained data is further processed and formalized to represent information and knowledge. Through the installation of a Knowledge Base, one is able to present the knowledge and to utilize it in product development systems, as to improve the product development process of future projects (Harris, 2009).

As mentioned, a well structured knowledge base is only the starting point for the utilization of project know-how in future development initiatives. In order to properly support designers and engineers, the proper KM systems must be implemented.

C.1.3 Knowledge Management Tools

Within the literature, a large number of tools to support the knowledge management process have been proposed. Cooper categorizes the tools as follows (Cooper, 2006):

- Authoring tools assist in the creation of text-based products.
- Collaboration tools facilitate the storage and retrieval of documents and support version and change control.
- Design rationale tools propose structure and capture rationale for design by proposing information during design process.

- Design tools assist in creation, visualization and analysis and simulation of proposed designs for various products.
- Groupware systems help the team and its members communicate and coordinate their work packages.
- Knowledge resources provide support for the acquisition, development, distribution and access to knowledge resources and specialized content.
- Knowledge-Based Systems and Decision Support Systems utilize data and analysis tools to provide improved decision input on easy-to-use interface; attempt to understand and codify implicit human knowledge in support of decision maker.

As it is the aim of this work to support the Engineering Change process, which is highly complex and dependent on a large number of input variables, Knowledge-Based Systems (KBS) seem to be the most suitable candidate for improving decision making support in the field of Engineering Change Management.

C.2 Types of DSS systems

In the scientific community, many different types of decision support systems have been proposed, yet they nearly all share a set of common traits as being „an interactive, flexible, and adaptable computer-based information system, especially developed for supporting the solution of a non-structured management problem for improved decision making. It utilizes data, provides an easy-to-use interface, and allows for the decision maker's own insights" (Turban, 1995).

Within the literature, a distinction between the various types of DSS is undertaken along the historical progression of research in the field and many authors adopt a categorization of the following types (Power, 2008):

1. Model-Driven DSS: This type of system uses limited data and parameters provided by decision makers to represent simple quantitative models. They are used for the access and manipulation of input and the simulation of variables. While large databases are not necessary for this kind of system, the limitations of Model-Driven DSS become evident for applications in highly complex environments with many input factors.
2. Data-Driven DSS: The most essential functionality of this type of system is the access and manipulation of a time series of data. With the rise of data warehouse systems, the analysis of data by computerized tools has gained more general applicability by tailoring online analytical processing for various decision contexts.
3. Communications-Driven DSS: This type of system is also often referred to as Group Decision Support System and uses network and communications technologies to facilitate decision-relevant

communication and cooperation. The dominant components are communication technologies, such as groupware, computer-based bulletin-boards and video conferencing tools.

4. Document-Driven DSS: Using computer storage and processing technologies to provide document retrieval and analysis, this type of systems can include a wide variety of datasets, such as images, sounds, videos, hypertext documents or even scanned documents. With the help of a search engine, the users can access the historical documents and product-related information and use them for the cases at hand.
5. Knowledge-Driven DSS: Being able to recommend actions to managers is the key feature of this kind of system. With the help of specialized knowledge about a domain, these systems can provide valuable inputs to decision makers. Artificial Intelligence (AI) is closely related to this type of DSS, as suggestions are derived by the system itself, using machine learning and computer reasoning.

In the field of Engineering Change Management, many interdependencies, explicit and implicit, are to be considered. Also, the input from experts is essential for the successful undertaking of a change initiative. Thus, in order to improve the decision making performance, a system needs to be readily available, even when human experts may not be, while also assisting the decision makers in reaching higher quality decisions by increasing consistency.

Considering the alternative types of DSS, a Knowledge-Driven DSS is determined to be the best match for the case of decision support in Engineering Change Management. In the following sections, the general structure and the system proposed by Niknam et al. will be described (Niknam, et al., 2014) as a foundation for the further development.

C.3 Structure of Knowledge-Based DSS

The core functionality of a Knowledge-Based Decision Support System (KBDSS) is its ability to simulate the human reasoning and to obtain recommendations for specific problems by heuristic reasoning. In achieving this, the system is highly dependent on the suitable design and setup of its components, which have been named as (Power, 2000):

1. Knowledge base: Within the context of a KBDSS, the knowledge base represents the part of the architecture in which the data is stored. It is important to point out, that this component is not merely a list of data points, but a much more extensive collection of expertise and knowledge in the form of facts, as well as rules.
2. Inference engine: This part draws upon the knowledge stored in the knowledge base and draws conclusions, based on the expertise represented as facts and rules, as well as the input on the case at hand.

3. User interface: Allowing for an interaction between the system and its users, the interface of a KBDSS must, apart from allowing further input for an enhancement of the knowledge base, provide an easy-to-use application for obtaining decision support on complex case instances.
4. Model of decision problem and conditions: In order to make inferences about a change case, the problem must be properly modeled with the relevant factors. This can be done using a variety of different user interfaces.

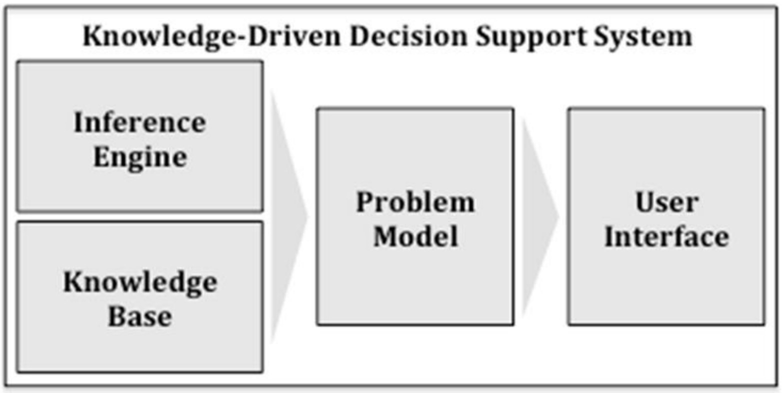

Figure 58 – Struture of a Knowledge-Based Decision Support System – Based on (Power, 2000)

Within the literature, this type of system is also often referred to as „expert system", as it can perform tasks, which normally require the input from human experts. Containing the models of decisions conditions and solutions, the knowledge base is at the center of KBS. There are three important approaches for using the knowledge based system for decision support (Shaofen, et al., 2010):

1. Rule-Based Reasoning (RBR): Specialized domain knowledge is represented in the form of IF <preconditions> THEN <conclusions>. This approach is especially suitable for well structured decision problems.
2. Case-Based Reasoning (CBR): In many situations it is difficult to obtain a complete set of rules to cover all possible eventualities. Case-Based Reasoning applies old cases to new situations and can thereby handle even ill-structured and incomplete decision situations by adopting old cases to explain new situations.
3. Hybrid-Reasoning: There is an increasing interest in the combination of RBR and CBR in recent years. Especially with the development of Bayesian Belief Networks (BBN), which are probabilistic inference engines that can also be used to reason under uncertainty, a wide range of complex semi-structured and unstructured decision problems become accessible.

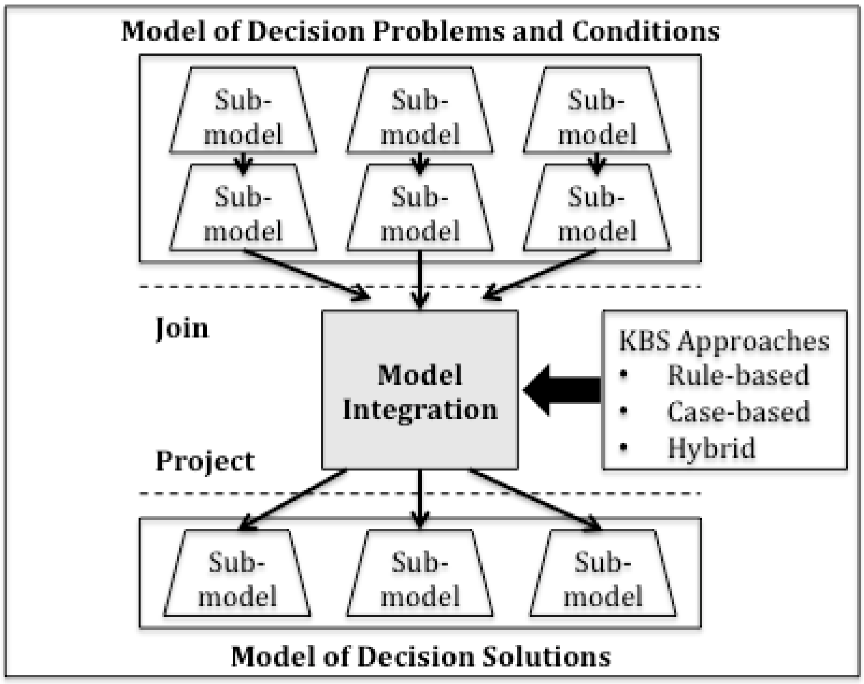

Figure 59 – Knowledge Based approaches to facilitate model integration – Based on (Shaofen, et al., 2010)

In summary, a KBS extracts a model of the current decision problem and its conditions and joins these facts with the knowledge stored in the system. This knowledge is applied in the form of Rule-Based Reasoning, Case-Based Reasoning or Hybrid Reasoning to project solutions for the decision case. In order to accomodate the mentioned functionality, a KBDSS must therefore integrate a wide range of data and models to solve a given problem (Zhou, et al., 2009). It is therefore imperative to properly structure and define the most essential part of such a system, namely the Knowledge Base.

Appendix D. Knowledge-Base Structure and Case Representation

D.1 Ontologies

Within the context of this work, ontologies play an integral role in enriching the data on Engineering Change cases with the domain knowledge on the structure and hierarchy of features. Therefore, an in-depth examination of the origin of the term and a formal definition are undertaken and provide the basis for the definition of the Engineering Change cases.

D.1.1 From Philosophical to Computer Science Ontologies

In today's society, a large amount of information is shared between various actors, including people and applications. Only by using uniformly characterized terminology can this information be transfered so that, in similar contexts, everyone understands the meaning and is able to grasp the concepts. One of the methods satisfying this need for „common understanding“ of concepts is the creation of ontologies (Sanchez, et al., 2007).

The term ontology had a long history in Philosophy before being used in Computer Science and took on a different meaning in this process. Aristotle was one of the first philosophers to wonder „What is being?“ in his Metaphysics. He came to the conclusion that all beings in the world must have some „thing“ or characteristic, which describes the property of „being“ to objects. One of the distinctions he draws is between principle and essence. While the principle refers to the „source point of something“, the essence is the „intrinsic reason of existence of being“ (Aristotle, 2006).

Focusing on the nature and structure of things per se, this approach to Ontologies is independent of further considerations and even on their actual existence (Guarino, et al., 2009). This is in contrast to the use of ontologies in the experimental sciences, like Computer Science, where discovery and modeling of reality under a certain perspective are the main goals. Three areas of research within Computer Science, namely Artificial Intelligence, Software Engineering and Database Development, have independently identified the need for knowledge representation in a structured way (Sanchez, et al., 2007):

- Within Artificial Intelligence, where the goal is to have agents reach decisions in an autonomous and systematic way, knowledge must be represented in a computer environment and be made accessible to automatic interpretation.
- In order to give a high level representation of a problem domain without considering implementation issues, the Database community also turned to abstract models.

- The basic paradigm of object-oriented programing has given a new style to representing elements involved in a Software Engineering problem. The two basic characteristics of each element are attributes (or properties) and methods (which allow for actions of the object) (Booch, 1993). Within this hierarchical way of thinking about the world, one object inherits properties and methods from its parents and thus allows for problem simplification and increased interoperability.

It becomes evident that all three domains have a need for solving the same problem of representing abstract concepts. This codified representation is the starting point for generating knowledge. However, each of the disciplines addresses this issue in a different manner, as each is interested in a specific problem and domain. Therefore, within the literature, there is a large number of definitions of the term ontology.

D.1.2 Formal definition of Ontologies

Most commonly adapted is the notion proposed by Gruber, who considers an ontology to be „an explicit specification of a conceptualization" (Gruber, 1993). In order to properly understand this notion, it is important to distinguish between the terms conceptualization and representation.

Conceptualization:

Conceptualization is the process by which the human mind forms a mental representation about a part of the reality. It is a simplified view of the world we want to understand. In developing this mental model, only essential characteristics of the element are considered and accidental properties are omitted (Sanchez, et al., 2007).

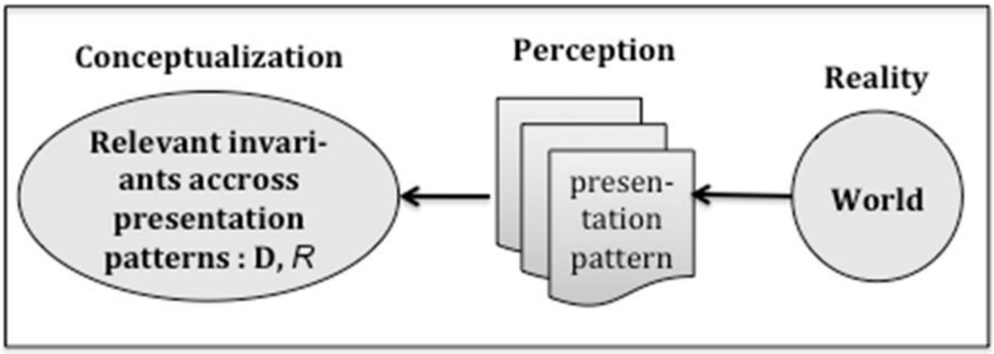

Figure 60 – Conceptualization – Based on (Guarino, et al., 2009)

Within the context of this work, we will follow the definition by Gensereth and Nilsson: „A body of formally represented knowledge is based on a conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly." (Gensereth & Nilsson, 1987).

Despite the very complex notion of conceptualization, we will follow Guarino et al. in their clear representation approach with the mathematical structure of an extensional relational structure (Guarino, et al., 2009):

Definition 1 – Extensional Relational Structure

An extensional relational structure is a tuple (D, R) where:

- *D is a set called the universe of discourse*
- *R is a set of relations on D*

In the above definition, members of the set R are ordinary mathematical relations on D, which can be represented as sets of ordered tuples of elements on D. Each element of R is therefore an extensional relation reflecting a specific world state of elements contained within D. In order to further specify the model, a formal description of the world concept is given (Guarino, et al., 2009):

Definition 2 – World

Regarding a specific system S we want to model, a world state for S is a maximal observable state of affairs. The system can therefore be characterized by a unique assignment of values to all the observable variables. A world is therefore a totally ordered set of world states, which correspond to the system's evolution in time. In abstracting from time, a world state simply coincides with a world.

Building on the terminology by Guarino et al., the intensional relations will subsequently be described in more detail (Guarino & Giaretta, 1995):

Definition 3 – Intensual, Conceptual Relation

Let S be an arbitrary system, D an arbitrary set of distinguished elements of S, and W the set of world states for S. Then we call the tuple <D,W> a domain space for S. This tuple fixes the space of variability within the universe of discourse D with respect to the states S can potentially take. Thus, an intensional relation of arity n on a domain D is a function from a set W of possible worlds on the set 2^{D^n} of all possible n-ary relations on D: $W \rightarrow 2^{D^n}$.

Having clarified the conceptual relations, the representation of the conceptualization must be defined. This will be the last element enabling us to move further within the ontology definition to the step of explicit specification.

Definition 4 – Intensional relational structure

The triple C = (D, W, $\mathfrak{R}$) represents an intensional relational structure with:

- *D as a universe of discourse*
- *W as a set of possible worlds*
- *𝔑 as a set of conceptual relations on the domain space <D, W>*

Formal Specification:

In any human communication, as well as in practical applications of the previously introduced conceptualization, it is imperative to use language to refer to the elements we want to describe. Therefore, we need to introduce a specific symbol, namely a predicate symbol, which represents the conceptual relation. This process is referred to in the literature as a language „committing to a conceptualization" (Guarino, et al., 2009).

The challenge is to assure that a given conceptualization is interpreted in the same way and that only those models which are intended according to the conceptualization are admitted. One could in principle explicitly specify a conceptualization intensionally or extensionally. The problem with an extensional specification is the fact that it requires listing the extensions of every conceptual relation for all possible worlds. This seems hardly feasible in many domains of discourse. Therefore, a more effective way of specifying a conceptualization is to fix a language to be used and to constrain the interpretations of such a language in an intensioinal way by introducing suitable axioms (Guarino, et al., 2009). The result is an approximate specification of a conceptualization, which is supported by the use of a given informal or formal language L.

Definition 5 – Extensional first-order structure

Be L a first-order logical language with the vocabulary V and S = (D, R) an extensional relational structure. An extensional first order structure (a model) is a tuple M = (S, I), where I is a total function $I : V \rightarrow D \cup R$ that maps each vocabulary symbol of V to an element of D or to an extensional relation belonging to the set R.

Definition 6 – Intensional first-order structure

Be L a first-order logical language with vocabulary V and C = (D, W, $\mathfrak{R}$) an intensional relational structure. An intensional first-order structure, called the ontological commitment, for L is a tuple K = (C, $\mathfrak{I}$), where $\mathfrak{I}$ is a total function $\mathfrak{I} : V \rightarrow D \cup \mathfrak{R}$ that maps each vocabulary symbol of V to either an element of D or an intensional relation belonging to the set $\mathfrak{R}$.

The ontological commitment is to be seen as an extension of the notion of model. While a model is an extensional account of meaning, the ontological commitment is an intensional account of meaning. In order to clarify the relationship between the two, we must introduce the notion of an intended model with respect to a certain ontological commitment (Guarino, et al., 2009):

Definition 7 – Intended Models

Be C = (D, W, $\mathfrak{R}$) a conceptualization, L a first-order logical language with vocabulary V and ontological commitment K = (C, $\mathfrak{I}$). We call a model M = (S, I), with S = (D, R), an intended model of L according to K, if:

1. *For all constant symbols C ∈ V we have I(c) = $\mathfrak{I}$(c)*
2. *A world w ∈ W exists, such that, for each predicate symbol v ∈ V there exists an intensional relation ρ ∈ $\mathfrak{R}$ such that $\mathfrak{I}$(v) = ρ and I(v) = ρ(w).*

The set $I_K(L)$ of all the models of L that are compatible with K is called the set of intended models of L according to K.

Having established the notion of intended models, the role of an ontology, which is considered to be a logical theory designed to account for the intended meaning of the vocabulary used by a logical example, can finally be clarified (Guarino, et al., 2009):

Definition 8 – Ontology

Be C a conceptualization, and L a logical language with vocabulary V and ontological commitment K. An ontology O_K for C with vocabulary V and ontological commitment K is a logical theory, which consists of a set of formulas of L, is designed so that the set of its models approximate as well as possible the set of intended models of L according to K.

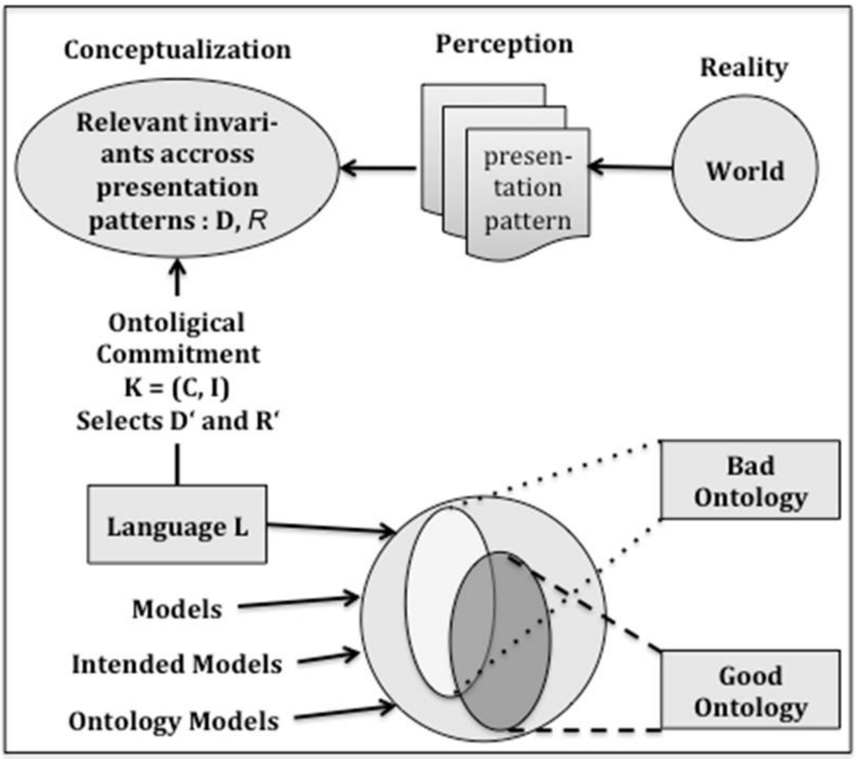

Figure 61 – Relationship between Phenomena in Reality and Ontologies (Guarino, et al., 2009)

It becomes evident that the definitions given above only capture the formal aspects of designing ontologies. Therefore, it is essential to clarify the criteria that determine a good ontology in more detail and to give pointers on what the vocabulary used in a wide range of different types of ontologies should be.

D.1.3 Design Criteria and Types of Ontologies

The previous definitions leading up to the formal definition of ontologies give few pointers to what set of principles to follow and what the design criteria for a good ontology should be. Therefore, roughly following Gomez Perez and Benjamins, a number of guidelines for the successful design of ontologies will be given (Gomez Perez & Benjamins, 1999):

- Clarity and Objectivity: An ontology should provide the meaning of the defined terms by the provision of objective definitions and documentation of natural language.
- Completeness: A definition expressed in terms of necessary and sufficient conditions is to be preferred over partial definitions.
- Coherence: In order to be able to draw correct inferences, a coherent structure and definition is required.
- Maximum Monotonic Extendability: New generalized and special terms of the ontology should be included in a way that does not require a revision of the existing terminology and definitions.
- Minimal Ontological Commitments: As few claims as possible about the world being modeled are to be given, in order to allow parties committed to the ontology the freedom to specialize and instantiate the ontology as they see fit.
- Ontological Distinction Principle: Classes within an ontology should be disjoint from each other.
- Diversification of Hierarchies: Increasing the power provided by multiple inheritance mechanisms, hierarchies can be distinguished.
- Modularity: Coupling between the modules is minimized.
- Minimization of the Semantic Distance between Sibling Concepts: Similar concepts are grouped and represented by use of the same primitives.
- Standardization: Names are to be standardized whenever possible to increase understanding clarity.

In recent years, a large number of ontologies have already been developed. Many are accessible within the public domain, yet a large portion of the detailed ontologies developed by companies are not available. Therefore, in the context of a Knowledge-Based Decision Support System, the ontological descriptions that are already established within an organization should be used to best describe the domain knowledge of the individual decision situation. The following paragraph will give a short overview of the types of ontologies most commonly used (Gomez Perez & Benjamins, 1999):

- **Knowledge Representation ontologies** capture the representation primitives which are used to formalize knowledge in knowledge representation paradigms.
- **General / Common ontologies** include vocabulary for the description of things, events, time, space, causality, behavior, function, etc.
- **Top-Level ontologies** are used to provide general notions under which all the terms within existing ontologies can be related.
- **Meta ontologies** can be reused across various domains and are therefore also referred to as Generic Ontologies.
- **Domain ontologies** provide vocabularies about the concepts described within a domain, as well as the relationships between concepts and the activities that take place within the certain domain. Thus, domain ontologies describe the theories and elementary principles governing that domain.
- **Task ontologies** describe the terms used to solve problems associated with tasks that may or may not be from the same domain with a systematic vocabulary, including generic names, generic verbs, generic adjectives and others used for the scheduling of tasks.

One might assume that an ideal ontology is achieved, when the developed models exactly coincide with the intended ones. Yet, this is many times not sufficient if either the vocabulary or the described domain of discourse are not properly chosen and defined. This is due to the distinction between the logical notion of a model and the ontological notion of a possible world. While the former is a combination of assignments of abstract relational structures to vocabulary elements, the latter is the combination of observed states of affairs within the system in question (Guarino, et al., 2009).

Therefore, it is essential to properly match the richness of the domain of discourse and the richness of the vocabulary to be used. Furthermore, for an ontology to properly describe a conceptualization, one needs to choose the right axiomatization, which is dependent on the language expressiveness.

D.1.4 Language and Language Expressiveness

Within theory and practice, a whole range of approaches for a language L exist. The spectrum ranges from more informal approaches, which only allow the definiiton of terms, with limited ability to assign meaning, to formal approaches, which are founded on clearly defined logical theories.

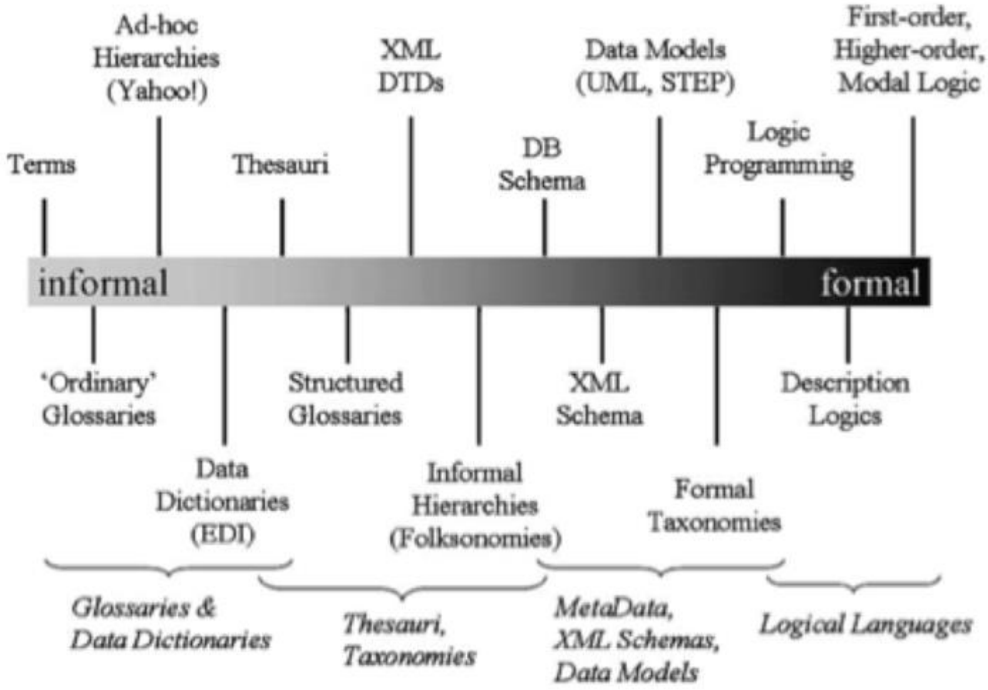

Figure 62 – Different approaches to the language L – Based on (Uschold, 2004)

Moving to the right of the proposed continuum, the amount of meaning that can be specified and the degree of formality increase, thereby reducing the ambiguity contained within the language. One can not draw a clear line on where a language L can be specified to be formal, yet it is important to note that there is a tradeoff between the expressiveness and efficiency when choosing a language L. While the full first-order logic, higher-order logic, or modal logic are very expressive, they do often not allow for sound and complete reasoning (Guarino, et al., 2009).

One of the most common languages used in the Semantic Web is the Resource Description Framework Schema (RDFS), which is a schema-specification language. An annotation with RDF consists of a set of statements, where each specifies the property of a resource (RDF, 1999) (Melnik & Decker, 2000). The advantage of using this language is its easy integration with existing Semantic Web technologies. Yet, it is not the most widely adopted within companies engaged in Engineering Change.

Within the production industry, the Standard for the Exchange of Product Model Data (STEP) has increasingly become the standard model. It is a norming series, which was developed within the frame of the International Organization for Standardization (ISO) and is therefore referred to as ISO 10303. Developed for the capturing and exchange of product information generated over the entire life-cycle, STEP consists of several Application Protocols (APs). Each of the AP contains the data models for describing product data for a defined family of products or a stage in its lifecycle. (Owen, 1993).

The STEP information models are interpreted using the EXPRESS modeling language. In contrast to most other modeling languages, EXPRESS posesses both, a graphical, as well as a textual notation. EXPRESS-G is the graphical representation and is a part of the textual EXPRESS. The most central element in this representation is the entity, which represents objects of the real world. The entities are further described by attributes, which can be of the following types (Anderl & Trippner, 2000):

Type	Values	Example
STRING	Chain of characters of arbitrary length and content	Name of product, Title of people
INTEGER	Full numbers in the range of (-65536,65536)	Version number, Year of market introduction
REAL	Rational numbers	Mathematical constants
NUMBER	Depending on the context either INTEGER or REAL	Prices of articles, Geometries
BOOLEAN	TRUE or FALSE	Is_Emergent_Change
LOGICAL	TRUE, FALSE, or UNKNOWN	Is_Emergent_Change
BINARY	0 or 1	Encrypted data, Images

Table 10 – Attribute Data Types in EXPRESS

One of the key features of the EXPRESS language is the ability to define super- and subtypes. There the concept of abstraction is used and a hierarchical relationship can be established, where certain properties and terms are inherited from the parent. This is essential for the definition of ontologies. Furthermore, the ability to express relations, as well as local and global rules, is important for the development of our case representations as ontologies (Anderl & Trippner, 2000).

In order to integrate the models within a STEP file into other applications, a method for exchanging information between different systems is required. For achieving an integration with, e.g., a JAVA application, which have become very common within the Computer Engineering domain, several approaches have been developed. Most prominent is the Common Object Request Broker Architecture (COBRA), which is a transparent component and enables and controls the communication with other applications (Anderl & Trippner, 2000). The object-oriented techniques are well suited for the integration of STEP data into a wider context and allow for the development of a Knowledge-Based Decision Support System on the basis of a rich set of product life-cycle data.

Taking into consideration the expressibility of the EXPRESS modeling language and the vocabulary already described within the STEP framework, it appears most feasible within the domain of Engineering Change Management to model the cases with this already well established methodology.

D.2 Case Representation

Having a sufficient amount of life-cycle related product and change process data is essential for the successful change impact evaluation based on previous Engineering Change initiatives. Therefore, the use of a STEP-compliant data model is recommended.

D.2.1 STEP compliance

Distributed over different STEP APs are information models relevant for representing Engineering Change initiatives. Most relevant are:

- Manufacturing: AP 224 (ISO/IS, 2000), AP 240 (ISO/IS, 2005)
- Systems Engineering: AP 233 (ISO/IS, 2003)
- Lifecycle: AP239 (ISO/IS, 2005)
- Product Data Management: AP 203 (ISO/IS, 2001), AP 212 (ISO/IS, 2001), AP 214 (ISO/IS, 2003), AP 232 (ISO/IS, 2002)

Within the manufacturing AP 240, one can find a number of entities relevant to the Engineering Change process, including Design_exception_notice, Engineering_change_proposal, and Engineering_change_order (ISO/IS, 2005).

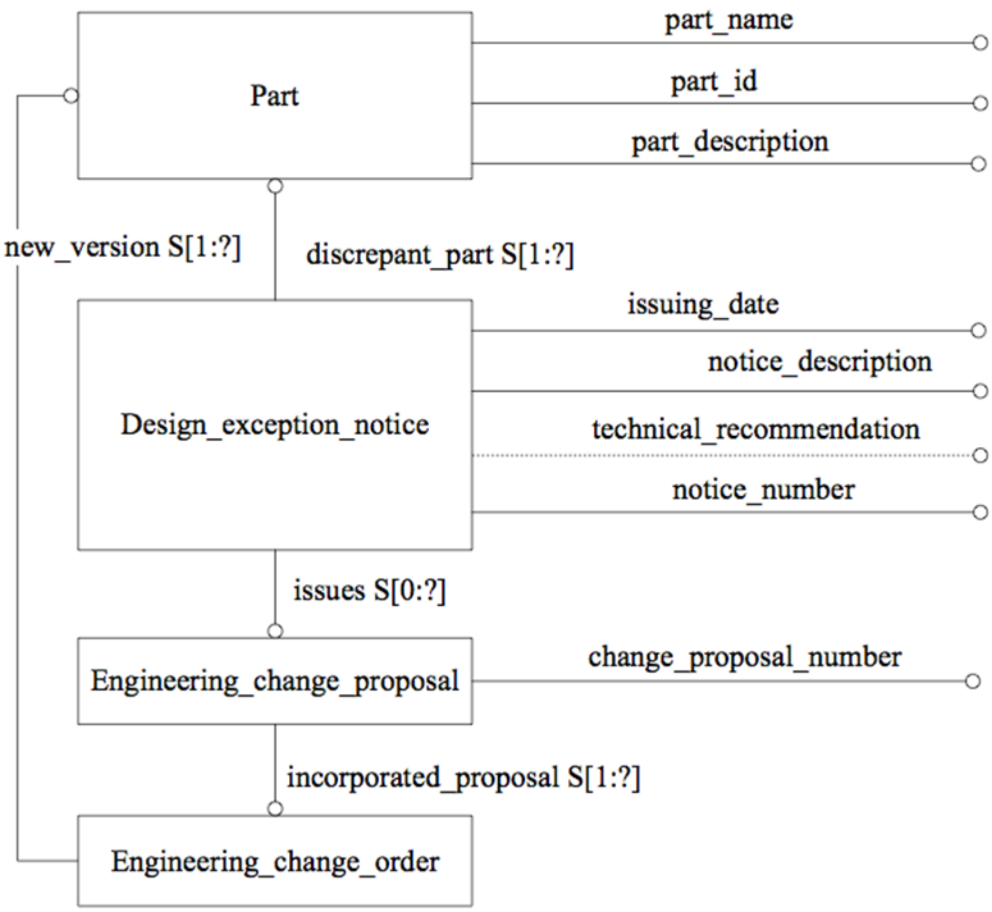

Figure 63 –EXPRESS-G illustration of core EC-related entities and attributes in AP 240 – From (Mehta, 2010)

In this illustration, established in EXPRESS-G, the entity Design_exception_notice represents a notification for an identified design discrepancy in creating the process plans for a given part such that there needs to be a technical recommendation made to correct the problem, before the process planning can continue. Further defining the Design_exception_notice is a subsequent Engineering_change_proposal entity, which describes possible amendments to a part leading up to an Engineering_change_order (Mehta, 2010).

Currently, STEP does not support the representation of all data associated with an Engineering Change case. Nonetheless, fundamental data structures are provided, which can be extended to capture the data required for describing the required EC data.

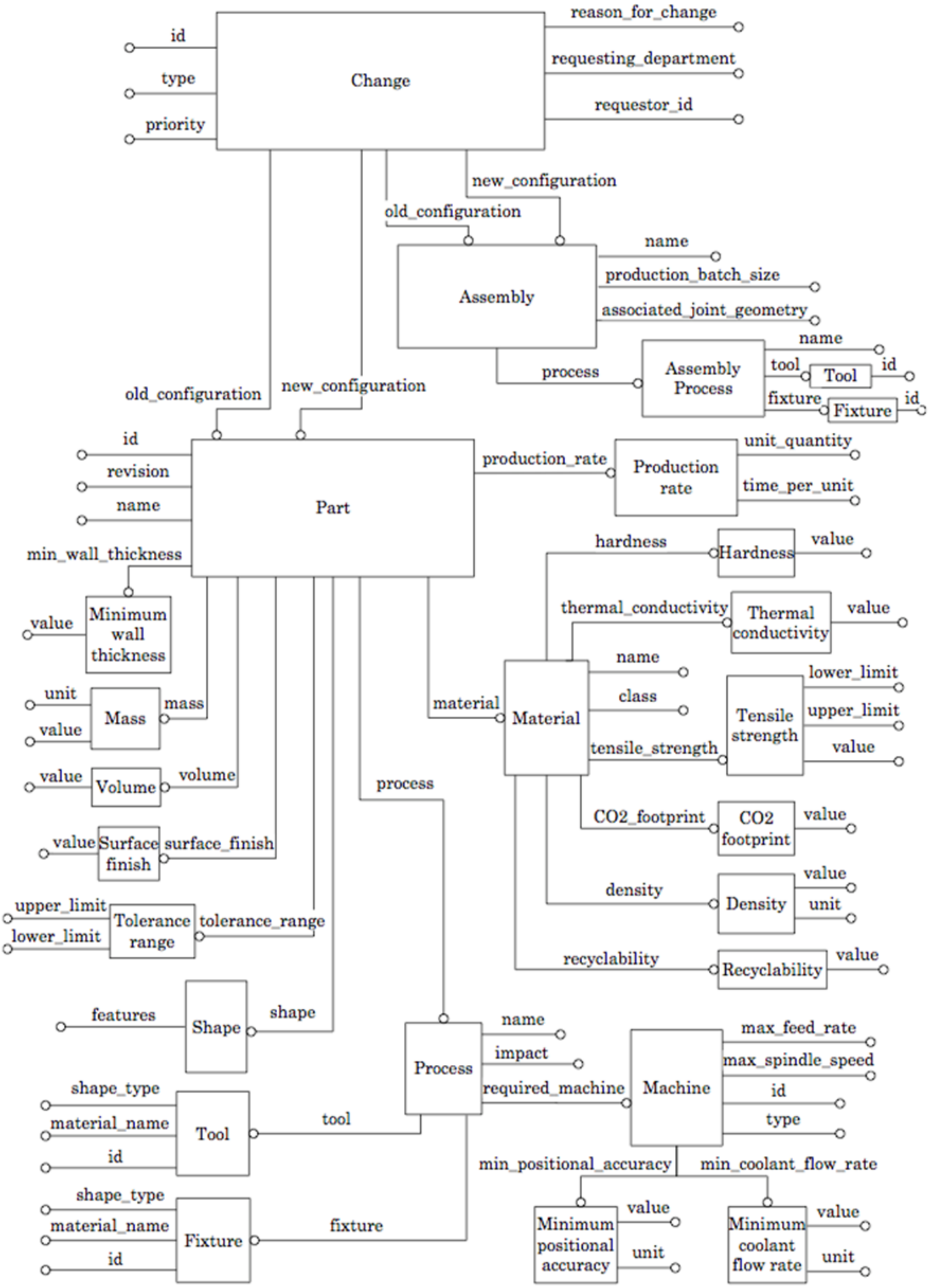

Figure 64 – Candidate STEP compliant Data Model of EC Knowledge – From (Mehta, 2010)

The preceding example, based on the work of Mehta, utilizes many elements associated with the STEP manufacturing AP 224 (ISO/IS, 2000) and AP 240 (ISO/IS, 2005), as well as a range of further concepts, which have been implemented to be able to properly describe an Engineering Change case (Mehta, 2010). It is at this point especially important to note the fact that, through the use of the relation old_configuration, which describes the state before the change is introduced, and the relation new_configuration, describing the state following the implementation of the change initiative, it becomes possible to extract information regarding the impact of executing an Engineering Change initiative.

D.2.2 Components of a Case Description

Even though the individual elements of describing an Engineering Change case are highly dependent on the organization in which it occurs, it is nonetheless possible to provide a general structure, which will make the extraction of knowledge from previous cases feasible.

The two main components of an Engineering Change case are the description (describing the change case), and the solution (representing the impact of the change). In order to clarify what is meant by this, a formal definition is developed:

Definition 9 – Engineering Change Case

An Engineering Change Case is a two-tuple $C = (C_d, C_s)$, where:

- *C_d represents a set of features describing the problem $C_d = \{f_1, \dots, f_n, O_{n+1}, \dots, O_j\}$ of the case with:*
    - *$f_1, \dots, f_n$ describing the features 1, ... , i of numerical and boolean type of the case and*
    - *$O_{n+1}, \dots, O_j$ describing the hierarchical ontologies of the features n +1, ... , j with text string type of the case*
- *C_s represents a set of attributes that describes the solution or impact $C_s = \{s_1, \dots, s_n\}$ with s_i being the solution feature i of the case.*

Case C_i	
Input	
Problem	Perf.
Product	SUV
Component	Axle
Process	Proto.
Type	Adj.
Priority	Urg.S.
...	...
Output	
Cost	200
Lead time	+30
Safety issue	S1,S4
Process iss.	P2,P8
...	...

Figure 65 – Engineering Change Case Representation

Despite the fact that the features and attributes of Engineering Change cases are highly dependent on the organization, the available data and the input by industry experts, some suggestions on frequently mentioned concepts are given. First, exemplary features for describing the problem, which are to be contained in the tuple C_d are described in the following table:

Attribute	Definition	Description	Data type
Product	Type of product in which the change occurs	e.g. car	STRING
Component	Part or component that is to be changed	e.g. engine	STRING
Change Reason	Description of the trigger for the EC initiative	e.g. installation	STRING
Solution Approach	Type of solution suggested for addressing the issue	e.g. relocation of stub	STRING
Process Timing	Timing of the product adaptation within the lifecycle	e.g. pilot production	STRING
Priority of Change	Determines the urgency of implementing the change	e.g. very urgent	STRING
Production Method	Type of production used for manufacturing	e.g. pressing	STRING
Number of Interfaces	Counts interfaces the part or component has		INTEGER
Number of Production Steps	Counts steps from beginning of part production to end		INTEGER
Production Rate	Measures output per time		INTEGER
Change ID	Unique identification number for the case		INTEGER
Accuracy Tolerance	Determines upper and lower bounds for production accuracy		REAL
Emergent Change	Describes whether a change is emergent or not		BOOLEAN
IT Involvement	Indicates whether the change involves an adjustment in software		BOOLEAN

Table 11 – Candidate Description Features of an Engineering Change Case

Based on the features that describe an Engineering Change case, one needs to also investigate the impacts of these undertakings in order to give valuable support in the risk and impact assessment step of the Engineering Change process. Therefore, examplary impact attributes are provided in the table below:

Attribute	Definition	Data type
Implementation Cost Engineering	Describes the engineering cost required for implementing the change	REAL
Implementation Cost Machine Tools	Describes the cost for adjusting the machine tools for producing the part or component	REAL

Implementation Cost Equipment	Describes cost for additional equipment that needs to be bought for implementing the change	REAL
Implementation Cost Process Adjustment	Describes the cost for adjusting the production process to match the change requirements	REAL
Change in Production Rate	Measures increase or decrease of production rate over a unit of time	INTEGER
Change in Number of Interfaces	Measures increase or decrease of Interfaces between the part or component and its surroundings	INTEGER
Change Propagation to Components in same Subsystem	Indicates whether Change Propagation occurs within the subsystem	BOOLEAN
Change Propagation to Manufacturing System	Indicates whether Change Propagation occurs between the part or component and the Manufacturing System	BOOLEAN
Change Propagation to Components in other Subsystems	Indicates whether Change Propagation occurs between the part or component and parts in other subsystem	BOOLEAN
Requires Production Shutdown	Indicates whether the production line needs to be specially shut down to implement the change	BOOLEAN
Change in Lead Time	Measures the increase or decrease in lead time following the change	INTEGER
Safety Issues	Describes possible safety issues when implementing the change	STRING
Process Issues	Describes possible issues within the production process, which need to be handled	STRING

Table 12 – Candidate Solution Features of an Engineering Change Case

Having introduced a range of features describing the change problem C_d and a possible set of attributes resulting from the change C_s, it becomes evident that not all the concepts are already realized within the STEP framework and therefore require additional implementation if they are to be regarded in the change impact analysis. Linked to this issue is the fact that some firms may currently not have any data available on some of the features and are therefore limited in their ability to predict any change impact in the respective categories.

Definition 10 – Case Base

Be C_1 , ... , C_n a range of previous cases with description $C_{n,d}$ and solution $C_{n,s}$, then these cases make up a case base CB of Engineering Change cases ranging form 1 to n.

Following the definition of the concept of an Engineering Change case through the two-tuple C = (C_d, C_s) and a case base made up of a number of individual EC cases, is the assessment of whether any previous case is similar to the case at hand and can therefore provide valuable insights to the decision maker.

D.2.3 Case Similarity

In the described Knowledge-Based Decision Support System introduced by Niknam et al., the similarity assessment is undertaken within the rules layer (Niknam, et al., 2014). Within literature, there are numerous approaches to determine the similarity between two case instances, of which some will later be intoduced.

D.2.4 Formal Definition

For the following formal description, we will assume that any feature i described with the data type STRING is part of an Ontology O_i. Due to the diverse factors in C_d, a complex similarity measure is to be developed. At this point, a formal definition will be given, while the detailed workings of the similarity assessment and the associated algorithms will be introduced in the section on case retrieval.

Definition 11 – Case Similarity Measure

Be $C_a = (C_{a,d}, C_{a,s})$ and $C_b = (C_{b,d}, C_{b,s})$ two cases with:

- *$C_{a,d}$ being the description of case a in the way that $C_{a,d} = \{f_{a,1}, \ldots , f_{a,n}, O_{a,n+1}, \ldots , O_{a,j}\}$*
    - *with $f_{a,1}, \ldots , f_{a,n}$ describing the features 1, ... , i of numerical and boolean type of case 1*
    - *with $O_{a,n+1}, \ldots , O_{a,j}$ describing the hierarchichal ontologies of the features n +1, ... , j with text string type of case 1*
- *$C_{b,d}$ being the description of case 2*
    - *with $f_{b,1}, \ldots , f_{b,n}$ describing the features 1, ... , i of numerical and boolean type of case b*
    - *with $O_{b,n+1}, \ldots , O_{b,j}$ describing the hierarchichal ontologies of the features n +1, ... , j with text string type of case b.*

Be w_i the feature weight for each feature 1, ... j with $w_i \in [0, 1]$.

Then, $S_{a,b} = \sum_{i=1}^{n} w_i sim\ (f_{a,i}, f_{b,i}) + \sum_{i=n+1}^{j} w_i sim\ (O_{a,n}, O_{b,n}) \Big/ \sum_i w_i$ is the Case Similarity Measure between case a and case b with $S_{a,b} \in [0, 1]$, where $S_{a,b} = 0$ represents two entirely different cases and $S_{a,b} = 1$ represents the maximum similarity between two cases.

It is important to note that the Case Similarity Measure represents a Fuzzy Set rating based on the features and feature ontologies with the respective concept weights allows for the comparison of the widest possible range. The choice of this approach is due to the fact that the definition of Engineering Change cases can vary greatly between individual organizations.

As one can expect there to be a feature difference between the current Engineering Change case and the case represented within the case base one wishes to adapt to extrapolate a recommendation for the Change Impact Analysis, a tuple representing the adaptation knowledge requirements needs to be established. Contained therein need not only be the different features and their values, but also the context in which the case is set.

Definition 12 - Adaptation Knowledge Requirements

Be $AKR_{a,b} = (f_d , v_d , f_c)$ the tuple for the adaptation knowledge requirement between a current case a and the previous case b already represented within the case base with:

- *f_d being the features f_i or ontologies O_i in which the two cases differ and*
- *$v_d = (v_{a,d} , v_{b,d})$ being the values of the case instances a and be in the features f_i or ontologies O_i in which the two cases differ and*
- *f_c representing the context in which the adaptation knowledge is to be applied, namely those features f_i or ontologies O_i in which the two cases are the same.*

At this point, the involvement of Domain Experts is required to further define such concepts as the similarity between concepts and ontologies. It could for example be reasonable to accept a range of numerical values as entirely similar and set a fixed similarity measure for values ranging between predefined bounds. Furthermore, the development of domain ontologies requires detailed knowledge on the interrelations between and hierarchical structure of individual instances of an Ontology O_n .

Even though ontologies and their definition have previously been thoroughly introduced, it seems essential to give further pointers on how to properly establish domain ontologies, which are an essential element within this framework, in a separate section.

D.2.5 Domain Ontology Development

In an ideal case, an organization implementing a KB-DSS already posesses an ontological representation of all Engineering Change related concepts. Yet, in practice, this can hardly be assumed. Therefore, one can either adapt preexisting ontologies, or develop ontologies based on the available data and the structures of the individual organization.

There is a range of attempts to establish ontologies for many of the aspects related to ECM. For the description of physical objects, the PhysSys has been developed (Borst, et al., 1995). Furthermore, the Enterprise Ontology developed at the University of Edinburgh within the Enterprise Project, provides ontologies divided in the sections Activities, Organization, Strategy, and Marketing (Uschold, et al., 1998). There have even been attempts at developing ontologies for representing Engineering Change cases. The approach by Wang and Wan describes an EC case by using the four concepts Engineering Change Object, Engineering Change Process, Engineering Change Data and Engineering Change of Organization (Wang & Wan, 2013). Yet, the level of detail, as well as the type of concepts used, do not match the requirements for decision support in Engineering Change Management. Therefore, based on the data and structures within an individual organization, a number of domain specific ontologies need to be developed.

Within the literature, there is a large body of research on methodologies for developing ontologies. The goal of using such a structured approach is to obtain good results through following a set of steps, which are based on established best practices. At this point, a few of the most prominent concepts will be introduced, while readers with further interest are referred to the work by Gomez Perez and Benjamins, as well as the introduction to ontologies by Sanchez et al. (Gomez Perez & Benjamins, 1999), (Sanchez, et al., 2007).

The *Cyc* knowledge base contains a large quantity of common sense knowledge. Through the course of its development, a process framework has emerged, which divides the ontology establishment process in three basic tasks (Lenat & Guha, 1990):

1. Manual extraction of common sense knowledge
2. Knowledge coding with the aid of tools using the knowledge already stored in the Cyc knowledge base
3. Computer managed extraction from the established common sense knowledge base

The language that was used to implement this system was CycL and required two activities to specify the ontology:

1. Development of a knowledge representation and a top-level ontology, which contains the most abstract concepts
2. Representation of the knowledge for the different domains.

Most commonly cited is the *Methontology* approach by Gomez-Perez et al., which is inspired by methodologies from software development and is divided into eight tasks (Sanchez, et al., 2007), (Gomez Perez, et al., 2003):

1. Building of glossary of terms with natural language definition, synonyms and acronyms
2. Development of concept taxonomies for the classification of concepts
3. Establishment of ad hoc binary relation diagrams identifying ad hoc relationships between concepts of the ontology and the concepts of other ontologies

4. Building of concept dictionary containing all domain concepts, their relations, their instances and classes, as well as their instance attributes
5. Description of each ad hoc binary relation appearing in the binary relation diagram, resulting in an ad hoc binary relation table
6. Detailed description of each instance attribute appearing on the concept dictionary
7. Detailed description of each class attribute appearing on the concept dictionary
8. Description of each constant specifying information related to the knowledge domain

Taking into account the formal definition of ontologies from the previous section, together with the introduced methodologies for establishing ontologies, it is now possible to properly model Engineering Change cases. As there is tremenous expertise and organizational context required, the formalization of the case structure and the corresponding features and ontologies will be reserved for the implementation of a Knowledge-Based DSS.

On the basis of the similarity between a current case and a similar case from the case base, the tuple for Adaptation Knowledge Requirements has been established. In order to provide better insights on the estimated impacts of a current case, the previous case needs to be adapted to match the features and context of the change initiative in question.

D.2.6 Case Adaptation Rules

Building on the assumption that a similar previous Engineering Change case can provide helpful insights into a current case, as the results are at least comparable, the use of adaptation rules goes one step further. There, it is assumed that the difference between the features of individual cases has a high predictive value for explaining the difference in results, as long as certain criteria are met.

Furthermore, it is assumed that the more often one observes the same difference in features entailing similar changes in the results, the higher the validity, with which one can assume the relationship between an adjustment of features and the resulting change in outcome exists. Therefore, the following formal definition of adaptation rules will include a confidence measure, as well as other characteristics making it possible to select the rules most appropriate for a given Adaptation Knowledge Requirement (AKR).

Definition 13 – Case Adaptation Rule

Be R_{AKR} a Case Adaptation Rule $R_{AKR} = \{ ID, f_d , v_d , f_c , A (C_s), conf \}$ for a given Adaptation Knowledge Requirement $AKR_{a,b}$ between case $C_a = (C_{a,d} , C_{a,s})$, and case $C_b = (C_{b,d} , C_{b,s})$, with:

- *ID being a unique identifier*
- *f_d being the features f_i or ontologies O_i in which the two cases differ and*

- *$v_d = (v_{a,d}, v_{b,d})$ being the values of the case instances a and b in the features f_i or ontologies O_i in which the two cases differ and*
- *f_c representing the context in which the adaptation knowledge is to be applied, namely those features f_i or ontologies O_i in which the two cases are the same*
- *$A(C_s)$ being the solution change function with $A(C_s) = C_s'$ adapting the case solution C_s to C_s' with additive or symbolic functions, depending on the features*
- *conf being the confidence rating of the rule.*

In order to collect all the rules, which are generated by various means, the definition of a Rule Base is necessary. This also allows for the systematic search of rules at time of case adaptation.

Definition 14 – Rule Base

Be $R_{AKR,1}, \ldots, R_{AKR,n}$ all the rules generated within a system, then these rules form the Rule Base RB of the system.

With the formal definition of Case Adaptation Rules and the Rule Base, the structure of the Knowledge-Base is described in its entirety. In order to achieve this, ontologies needed to be introduced, as they serve as the core structure for the information model of Engineering Change cases. Subsequently, the individual layers of the Knowledge-Base were described.

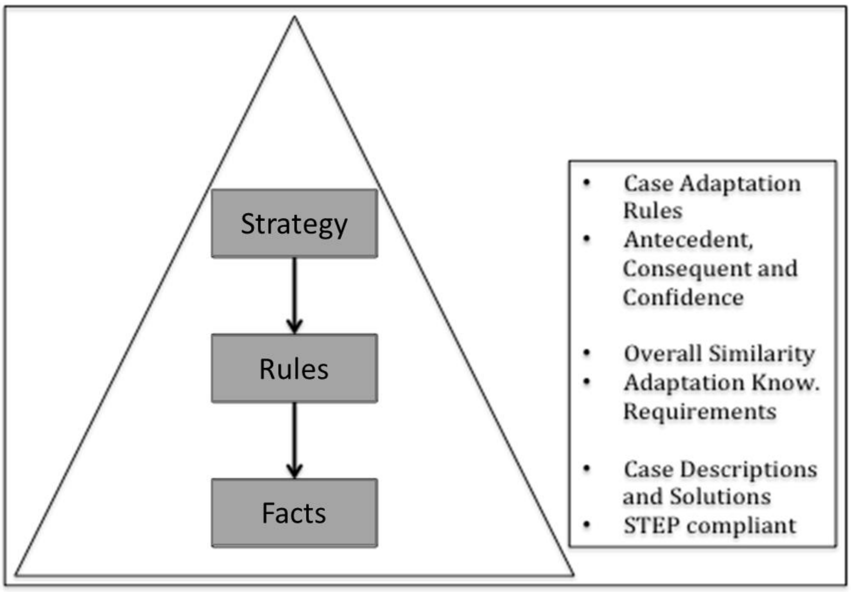

Figure 66 – Components of the Knowledge Base

The facts layer, which contains the previous cases, was characterized using a STEP compliant information model. Within the rules layer, case similarity was defined, based on the previously defined case structure. Furthermore, the concept of the Case Adaptation Requirement was introduced, which identifies not only the feature-differences, but also the context of a given pair of cases, which need to be adapted. Finally, Case Adaptation Rules were introduced, which integrate the antecedent, consequent and confidence features. On the basis of this tuple, the selection of the most suitable rule becomes feasible and enables a further enhancement of the predictive quality of the described Knowledge-Based Decision Support System.

It is important to note that the Knowledge-Base has deliberately been designed in a way that is non-specific to the way in which case similarity is calculated, or the way in which adaptation rules are generated and applied. This is due to the fact that the author intends to provide a reusable framework, describing the domain of Engineering Change Management, based on an ontological description in a STEP compliant information model.

This separation of the domain model from the Problem Solving Methods (PSM) allows for the reuse of the domain knowledge and enables a relatively low effort adjustment of PSM within the system. In the two following chapters, we will introduce algorithms for solving the problems of assessing Case Similarity and for the generation and application of Case Adaptation Rules.

Appendix E. Similarity Assessment and Case Retrieval

E.1 Similarity Assessment

One of the most important aspects of the proposed Knowledge-Based Decision Support System is the identification of previous cases, which are similar to the case at hand and can therefore provide helpful insights during the Change Impact Analysis. Within the literature, a large number of approaches to similarity assessment are proposed. Nonetheless, the recurring pattern is, as already described in the definition of the Similarity Measure, that a similarity measure for each feature is determined and then included in the overall assessment with a certain weight.

Explicitly focusing on Engineering Change management, Joshi suggests an approach, which assesses similarity of change cases based on only a few specific attribute values. The Issue Based Information System (IBIS) is developed to compute similarity between values of reason of change and a predefined look-up table is used to analyze the similarity of the other attributes. Following this, the n most similar changes are utilized (Joshi, 2007).

One of the more frequently cited approaches is that of Lee et al., who assess similarity of Engineering Change cases by weighing concept similarities of five ontologies, namely product, component, problem and solution (Lee, et al., 2006). Using Resnik's information-based measure, which automatically measures the similarity between instances of each ontology, even the depth of the ontology is considered (Resnik, 1999). The weights are determined using the Analytic Hierarchy Process (AHP), which uses expert judgement in pair-wise comparison of ontologies (Chen & Huang, 2001).

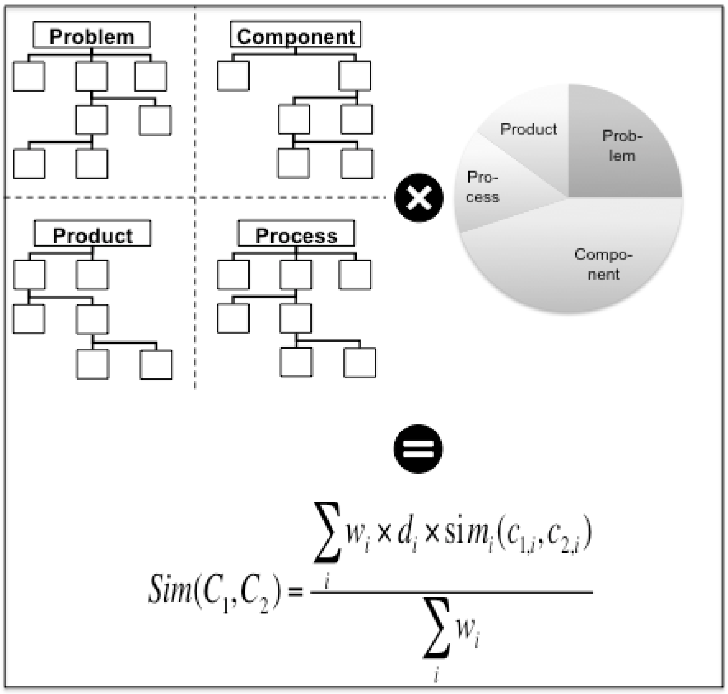

Figure 67 – Similarity Assessment – Based on (Lee, et al., 2006)

While this approach does not incorporate any differences in the values contained within the ontologies and requires a high amount of effort for determining the weights with the AHP, it is very flexible in incorporating various features and ontologies in the similarity measure and thereby allows for an adjustment of the assessment process based on the requirements of individual organizations.

Within the context of this work, the basic approach of Lee et al., which includes the use of concept similarity and feature weights determined by a multi criteria decision making method, namely the AHP, is utilized (Lee, et al., 2006). Yet, as the domain of ECM is complex and many features, even the ones represented in ontologies, can have different values within the concept, an evaluation of attribute similarity will be included. As there are various data types, the attribute similarity will differ between numerical, boolean and linguistic features.

The starting point for this approach is the definition of the similarity measure from the previous chapter, which states:

Formula 1 – Overall Case Similarity

$$S_{a,b} = \sum_{i=1}^{n} w_i sim\,(f_{a,i}, f_{b,i}) + \sum_{i=n+1}^{j} w_i sim\,(O_{a,n}, O_{b,n}) \Big/ \sum_i w_i$$

Based on this definition, it is necessary to define the individual components in order to make calculation of an Overall Case Similarity possible. When assessing the similarity of features, which are not represented in an ontology, attribute similarity is sufficient. In order to include the domain information already stored within the structure of an ontology, features represented in ontologies are further described using concept similarity.

Formula 2 – Attribute Similarity Boolean

$$sim\,(f_{a,i}\,, f_{b,i}\,) = \begin{cases} 1\; if\; v_{a,i} = v_{b,i} \\ 0\; if\; v_{a,i} \neq v_{b,i} \end{cases}$$

The similarity assessment of boolean features is very straightforward, yet when it comes to numerical variables, the approach needs to get more elaborate, as an interval of values could potentially be considered identical. This is important when dealing with noisy data, which, even if not perfectly identical, still expresses a negligable difference. There are various ways of dealing with this issue, which will be greatly dependent on the nature of the data. In this work, the range of possible feature values will be considered and divided into N intervals. A division into three intervals is, without loss of generality, proposed, as it allows for the classification into equivalent, comparable and different values to be categorized as such, while being able to handle the kind of noisy data one would expect within the field of Engineering Change Management.

Formula 3 – Attribute Similarity Numeric

$$sim\,(f_{a,i}\,, f_{b,i}\,) = \begin{cases} 1\; if\; 1 - \sqrt{\left(\frac{v_{a,i} - v_{b,i}}{\max(\,v_i\,) - \min(\,v_i\,)}\right)^2} \geq \frac{3}{4} \\ 0{,}5\; if\; 1 - \sqrt{\left(\frac{v_{a,i} - v_{b,i}}{\max(\,v_i\,) - \min(\,v_i\,)}\right)^2} \in \left[\frac{1}{2}\,, \quad \frac{3}{4}\right] \\ 0\; else \end{cases}$$

As far as linguistic feature values are concerned, it is important to note that a similarity between more than two instances requires an ordinal structure, which allows for a comparison of the difference between individual values (Belanche & Orozco, 2011). One such feature could be the urgency of an Engineering Change initiative, which can be brought into ascending order and therefore provide the basis for an attibute similarity measure.

Formula 4 – Attribute Similarity Linguistic

Be v_i the value of a linguistic feature with an ordinal structure $v_1 < v_2 < \ldots < v_n$, then

$$sim\,(\,f_{a,i}\,, f_{b,i}\,) = \begin{cases} 1\; if\; v_{a,i} = v_{b,i} \\ . \\ 1 - \frac{|j-k|}{n+1}\; where\; j\; is\; the\; ordinal\; value \\ of\; v_{a,i} = v_j\; and\; v_{b,i} = v_k \\ . \\ 0\; else \end{cases}$$

Having defined the Attribute Similarity measures for the different kinds of feature types, it is still necessary to assess Concept Similarity for the features represented in ontology structures. While in an ideal case, all features are described via ontologies, the field of Engineering Change Management requires, due to its complexity, a more complete definition. In the following, a Concept Similarity measure will be introduced, which, if necessary can be combined with the Attribute Similarity, enables a similarity assessment of complex features represented through hierarchical ontologies.

Formula 5 – Concept Similarity

$$sim\left(O_{a,n}, O_{b,n} \right) = -a_i \times \log \frac{N\left(\{C_i \mid O_s \in C_i\} \right)}{N\left(U \right)}$$

where O_s is the Closest Common Parent (CCP) to both, $O_{a,\ n}$ and $O_{b,\ n}$ within Ontology O_n and $N\left(\{C_i \mid O_s \in C_i\} \right)$ is the number of EC cases belonging to the concept O_s . Furthermore, N (U) is the number of EC cases within the case base and

$$a_n = \log \frac{\sum_m N\left(O_m \right)}{N\left(O_n \right)}$$

is the compensation factor for reducing size effects. This measure is similar to the Inverse Document Frequency (IDF) measure widely used in information retrieval (Salton & McGill, 1983) *and is therefore referred to as Inverse Concept Frequency (ICF)* (Lee, et al., 2006).

For the rare case that a concept represented by an ontology has differing values within the same concept, an Integrated Similarity measure can be applied, which is in line with Wang and Wan (Wang & Wan, 2013). Yet, this seems rather unlikely within the domain of ECM and is only mentioned for the sake of a complete framework covering all possible requirements on the system.

As previously mentioned, the Analytic Hierarchy Process is used to determine the weights of the features and ontologies describing the Engineering Change cases. The AHP is a structured technique used for making complex judgements and was first introduced by Saaty (Saaty & Vargas, 2012). The first step is to construct a (n x n) matrix A, where n is the number of features or ontologies to be weighed. Then, each expert sets $a_{i,\ j}$ as the preference of f_i or O_i over f_j or O_j when retrieving the relevant engineering change cases. The following rules are assumed for the elements in the matrix A:

1. If $a_{i,\ j} = \alpha$, then $a_{j,\ i} = {}^{1}/_{\alpha}, \alpha \neq 0$
2. If O_i is judged to be of an equal relative importance as O_j, then $a_{i,\ j} = 1$ and $a_{j,\ i} = 1$, which also implies $a_{i,\ i} = 1, \forall\, i$

The objective of obtaining the weight value w_i for calculating the overall similarity, the matrix A is used, in combination with the following formula (Lee, et al., 2006):

Formula 6 – Feature Weight

$$w_i = \frac{1}{\lambda_{max}} a_{i,j} w_j \text{ , } (i = 1, \dots , n)$$

where λ_{max} *is the maximum eigenvalue of matrix A* (Saaty, 1980).

On the basis of the similarity assessment, which considers various features of Engineering Change case, one can determine at which points further input in the form of adptation rules is required. In doing so, the most relevant cases, namely those with the highest similarity rating, are retrieved and the Adaptation Knowledge Requirement is derived as input for the adaptation stage.

E.2 Case Retrieval and Adaptation Knowledge Requirement

In order to draw conclusions from previous cases, one must determine, which instances are relevant to a current Engineering Change case. Based on the previously defined concepts of attibute and concept similarity, as well as on the feature weights derived via the Analytic Hierarchy Process, the calculation of an Overall Similarity Measure is achieved.

In the following algorithm, a structured approach to determining a ranking of the most similar cases is introduced, which are subsequently retrieved, reused and revised with the help of adaptation rules.

Definition 15 – Case Retrieval Algorithm

Input:

- *New Case* C_p *described with* $C_{p,d} = \{f_{p,1}, \dots , f_{p,n}, O_{p,n+1}, \dots , O_{p,j}\}$
- *A Case Base CB with m previous cases* C_m
- *Feature weights* w_i *for each concept*

Output:

- *Overall Case Similarity between the new case* C_p *and each case* C_m *contained within the Case Base CB*
- *Adaptation Knowledge Requirements for the q most relevant cases of the Case Base CB*

Step 1: *Calculate Overall Case Similarity:*

Starting from $k = 1$ *, For* $k \leq m$ *select one case* C_k

Starting from $i = 1$*, For* $i \leq n$ *select feature* f_i *Begin*

Comparison of $f_{p,i}$ *and* $f_{k,i}$ *:*

- *If f_i is boolean, use Formula 2 and obtain sim ($f_{p, i}$, $f_{k, i}$)*
- *If f_i is numeric, use Formula 3 and obtain sim ($f_{p, i}$, $f_{k, i}$)*
- *If f_i is linguistic, use Formula 4 and obtain sim ($f_{p, i}$, $f_{k, i}$)*

End

Starting from i = n+ 1, For i ≤ j Begin

Comparison of $O_{p, i}$ and $O_{k, i}$:

Use Formula 5 and obtain concept similarity sim ($O_{p, i}$, $O_{k, i}$)

End

Calculate Overall Case Similarity

Use Formula 1 with feature weights w_i to compute $S_{p, k}$

End

Step 2: *Retrieve q most similar cases and derive Adaptation Knowledge Requirements*

Order cases C_k in descending similarity order from $S_{p, k, 1}$ to $S_{p, k, m}$

Starting from $S_{p, k, 1}$, For all q most similar cases Begin

Starting from i = 1, For i ≤ j select a feature f_i or ontology O_i Begin

If sim ($f_{p, i}$, $f_{k, i}$) = 1, $f_i \in f_{k, c}$

Else $f_i \in f_{k, d}$ and ($v_{p, i}$, $v_{k, i}$) $\in v_{k, d}$

End

As can be seen from the previous description of the Case Retrieval Algorithm, which is held in pseudo-code for simplicity, a large number of operations take place in order to assess the Overall Case Similarity and retrieve the q most similar cases, together with the corresponding Adaptation Knowledge Requirements.

Within the context of this work, it appears feasible to retrieve a fixed number q of relevant cases, rather than introducing a threshold similarity measure, which is responsible for deciding on case selection. Yet, should a threshold seem more suitable at the time of implementation, the system is easily amended. This fact is a further advantage of separating the domain knowledge from the Problem Solving Methods.

In the course of the subsequent chapter, an approach to adaptation rule generation and application is intoduced. On the basis of the similarity measure proposed in this section, a further refinement of the result is initiated. As a result, the predictions for the change impact of the current case are improved.

Appendix F. Adaptation Rules

F.1 Types of Adaptation Models

With the many different models of case adaptation available, a variety of systems is imaginable and it is necessary to find a model appropriate to the complexity of the application domain. An overview of the basic types, based on the classification by Mitra and Basak is given in the following paragraphs (Mitra & Basak, 2005).

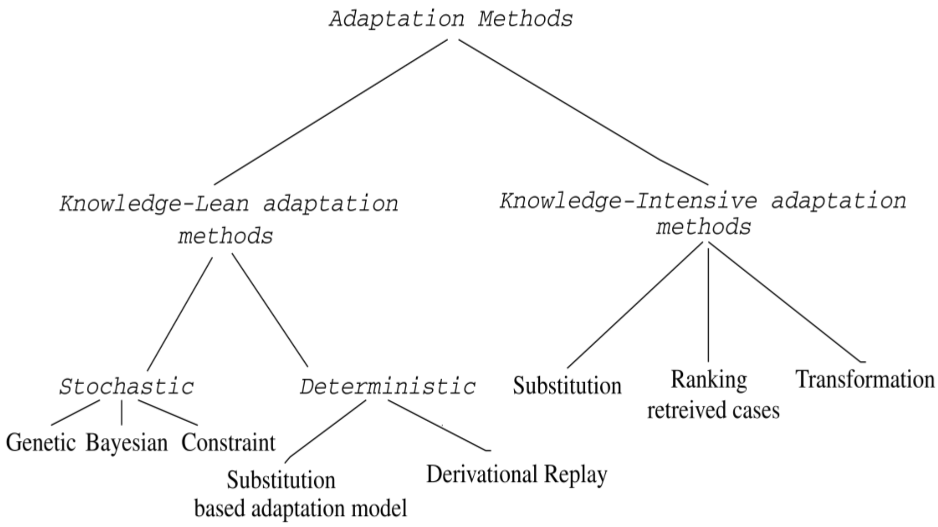

Figure 68 – Adaptation Methods classified by Domain Knowledge Requirement – From (Mitra & Basak, 2005)

The simplest kind of adaptation is null adaptation, which does not include any adaptation and uses the retrieved solution directly.

Within the area of transformational adaptation, usually a fixed set of adaptation operaters are used to modify the solution based on domain knowledge. Substitutional adaptation only changes the values of attributes, leaving the structure unchanged. The values of the previous case are used and amended using previously identified rules. During structural adaptation, more substantial modifications are undertaken. It is thus possible to add, remove or reorganize elements of the solution under conditions codified in a set of transformation rules.

The method of ranking retrieved cases uses a weighted average of the attribute values to derive a solution. This approach is of course only applicable in domains with numerical attribute values.

Generative adaptation works radically differently from transformational adaptation, which transfers a previous solution to the new problem. During generative adaptation, the derivation of a previous solution (reasoning trace) is transferred to the new case. These so called solution traces are then replayed in the context of the new problem to obtain a solution from scratch. Two different strategies are distinguished, namely one shot replay and interleaved replay. They basically only differ in the sequence in which replay occurs and not in the steps that are being replayed (Wilke & Bergmann, 1998). In both cases, a problem solver will identify which portions of the solution trace can be reused and apply them to the new situation.

Another method for solving case adaptation problems is to use adaptation guided by constraint satisfaction. There, a number of choices need to be made, represented by the variables. Each is associated with a number of options, named the variable domain, and a set or relations between the choices, named the constraints. In solving the problem, systems of this kind assign a value to each variable respecting all the problem constraints (Mitra & Basak, 2005). This type of model is especially well fit when determining change propagation within a solution set.

Based on a strong theoretical foundation is the approach using Bayesian Belief Networks (BBN), which is a probabilistic graphical model representing a set of random variables and their conditional dependencies. The solution to a given adaptation problem is obtained using conditional probabilities produced by the Bayesian case matching process and represents corrections to the original values of the input case, which is assumed to have a high predictive value for the case at hand (Liu, et al., 2010).

One of the more complex approaches is that of using genetic algorithms. There, the case base forms an initial population of genotypes and the algorithm retreives partial matching cases with special design requirements. In the following step, the retrieved cases are mapped into a genotype representation and crossover and mutation operators are applied. The final solution is obtained by matching the newly generated genotypes into the corresponding phenotypes (the cases) by inferring the values for the attributes and adding the context of the new design (Mitra & Basak, 2005). The parallel to the concept of genes is intentional, as the attributes used for the description of the case becomes a set of genes forming the genotype within the context of a genetic algorithm.

In addition to the previously introduced models, Artificial Neural Networks (ANN) deserve mention. These are computational models inspired by the structure of biological neural networks and are also represented by a system of interconnected „neurons“, which can compute values from the inputs they get. This type of system consists of one or more hidden layers with nonlinear activation functions, which carry out nonlinear transformations on the input patterns. It is thus difficult to explain the induction process, which represents the knowledge learned and the result of any given input case (Policastro, et al., 2006).

On the basis of the different models for adaptation, a number of different tools for adaptation have been developed. Some of these are subsequently introduced, before the sources of adaptation knowledge and the suitable application is discussed in detail.

F.2 Sources of Adaptation Knowledge and Rule Generation

Within this section, some approaches and the corresponding systems to generating rules are introduced. For a more comprehensive overview on a wide range of systems covering all the adaptation methods mentioned in the previous section, the work of Mitra and Basak is highly recommended (Mitra & Basak, 2005).

An ideal reasoning system should be able to learn adaptation knowledge automatically from a variety of sources with minimal knowledge-engineering required. Yet, most systems are either merely case retrieval systems, which do not use rules to adjust the solution to the case at hand, or require a lot of work by domain experts to provide the adaptation rules. Within the literature, the difficulty to obtain good adaptation rules has been referred to as the Adaptation Knowledge Bottleneck (Hanney & Keane, 1997).

A number of approaches have been proposed to close this gap by automating the Adaptation Knowldege Acquisition (AKA) through machine learning techniques and Knowledge Discovery in Databases (KDD). One of the most prominent examples is that of Leake et al., who introduce the concept of adaptation cases (Leake, et al., 1996). Within the domain of disaster response, they have developed the DIAL system, which uses given adaptation knowledge strategies to be applied, resulting in a specific set of adaptation cases. In this system, adaptation is considered a combination of search and transformation. Whenever differences between the current problem and the retrieved similar case arise and adaptation fails, suitable prior cases are found and rule-based adaptation is employed and the results stored as an adaptation case for future use (Leake, et al., 1995).

Another approach, which has been successfully tested within the domain of tablet formulation is that of Craw et al., who further extend the concept of adaptation cases. In their approach, a nearest-neighbor retrieval is followed by incremental adaptation based on the use of different learning algorithms (Craw, et al., 2006).

Frequently cited within the literature is the method by Anand et al., who use a knowledge-light approach (Anand, et al., 1998). The first step is to cluster the records within the database and acquire cases and adaptation knowledge with the use of the C4.5 algorithm, which is commonly used in machine learning. While the process is rather complex, it has the advantage of not being dependent on domain expert input (Lee, 2003).

One of the more complex systems is that by Policastro et al., who first use the k-NN algorithm to retrieve the most similar cases and employ a hybrid committee approach for case adaptation. They propose the use of three estimators, namely the Multi Layer Perceptron (MLP) neural network, the symbolic learning algorithm M5 and the Support Vector Machine (SVM) technique, and a combiner to perform the adaptation. MLP networks are an ANN model and are commonly used for pattern recognition. They present one or more hidden layers with nonlinear activation functions and carry out successive transformation on the input patterns. They are thus able to model highly complex relations of nonlinearly separable problems. The M5 symbolic algorithm generates models which are in the form of classification trees with corresponding regression equations. During the construction of the model tree, a divide-and-conquer approach is recursively applied to create new models. Then, a linear model is calculated for each inner node of the tree using a standard regression process, before pruning completes the set-up. The concept of Support Vector Machines is based on statistical learning theory and combines generalization control with a technique dealing with the dimensionality problem. Hyperplanes are used as decision surfaces and a maximization of the margins between positive and negative classes is undertaken. Through the use of kernels, even non-linear classification is possible, by mapping the inputs into higher dimensional feature space through the kernel functions. This represents a very efficient learning algorithm, which uses a set of training data to derive complex classifications and regressions. With the use of the mentioned machine learning algorithms, a data set of adaptation patterns is generated, which is used to train a committee of algorithms to automatically perform the case adaptation. (Policastro, et al., 2006)

Despite the author’s initial excitement about the potential of the previous approach, there are several limitations, which inhibit the application within the domain of Engineering Change Management. First, the system is based on the assumption that the Case Base is representative and includes all future problems, which can occur within the domain. This is obviously not the case for Engineering Change intitiatives, which vary greatly and consistently take on new problems not previously addressed. Furthermore, the introduced algorithms do not provide any explanation on what has been learned and make the solution process impossible to trace. This is not acceptable for decision makers in the domain of Engineering Change Management. Therefore, these elegant methods are not applicable for the proposed Knowledge-Based Decision Support System.

This work will be based on the seminal approach by Hanney and Keane, who have demonstrated the feasibility of retrieving adaptation rules from the Case Base, while including domain knowledge in the refinement of the Rule Base (Hanney & Keane, 1996). This approach is based on the assumption that the differences occuring between cases in the Case Base are representative of the differences that arise between the current problem and the Case Base (Hanney & Keane, 1997) (Smyth & Keane, 1995).

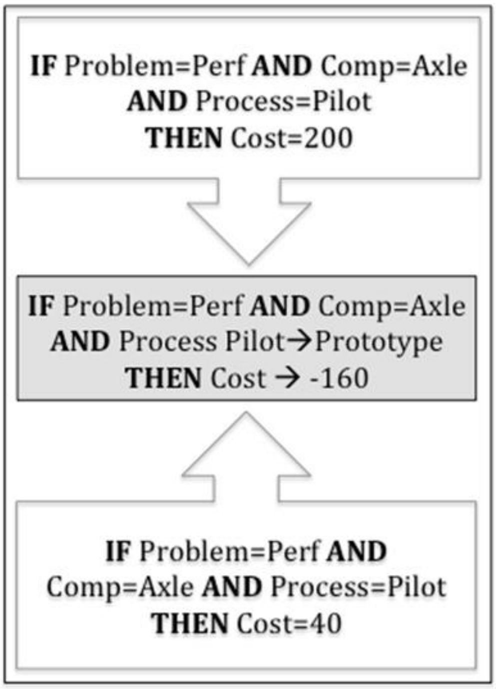

Figure 69 – Deriving Adaptation Rules from the Case Base

In this approach, a pairwise comparison of previous cases of the case base is undertaken. Based on the feature differences, adaptation rules are generated. One of the ways to go about mining such rules is by using association rule learning algorithms, like the Apriori algorithm, which is traditionally used in market basket analysis (Nahar, et al., 2013). The algorithm introduced in this work is based on a similar premise, yet it uses the Overall Similarity Measure and the Adaptation Knowledge Requirements to derive adaptation rules.

In the following, the traditional approach to rule generation using domain experts will be introduced. This was originally used in Expert Systems, which were among the first implementations of Knowledge-Based Systems and require a manual input of all feature differences that are likely to arise within a certain domain. Of course, this approach is very labor intensive and requires a deep understanding of the domain. Nonetheless, the rules generated in such a way are of extremely high confidence and are valuable in providing a framework for the subsequent generation of adaptation rules from the Case Base.

F.2.1 Using Domain Knowledge and Experts to Learn Adaptation Rules

Within the context of this work, expert knowledge is not only considered for the construction of the ontologies describing the case, but also for providing a base of known adaptation rules and determining under which context certain factors become irrelevant.

The known adaptation rules are considered to be more reliable than the rules later generated from within the case base and are therefore awarded with a confidence rating of conf = 1. This means that, in case a certain feature difference is covered by a known adaptation rule, no learned rule will be applied for this certain adaptation. The previously introduced structure of antecedent and consequent remains intact and is applied according to the Adaptation Rule Application algorithm later introduced.

When it comes to the implementation of rules describing the relevance of certain factors, the proposed model is able to delete those factors from rules which are determined to be irrelevant within a certain context. This is achieved by implementing rules k with confidence conf = 1, an empty adaptation function A (C) and the irrelevant factor $f_i = f_{k,d}$, together with the determining context factors contained within $f_{k,c}$. The proposed algorithm for learning adaptation rules from the Case Base will then omit these factors from the rules and thus achieve an enhanced reasoning capability based on domain knowledge present within the organization.

As can be seen, the manual input of these rules is very labor intensive. Nonetheless, in this way, it is possible to codify domain knowledge, which is not represented within the Case Base, but tacitly available through an organization's domain experts. Especially in long life-cycle industries, where experts may only be involved in parts of the implementation, this way of retaining knowledge is essential.

F.2.2 Learning Adaptation Rules from the Case Base

As desicribed at the beginning of this chapter, a number of different approaches to rule generation and application are proposed within the literature. Within the domain of Engineering Change Management, it is especially important for decision makers to be able to understand from where the assumptions for a proposed impact evaluation is coming. Therefore, the following algorithm will derive adaptation rules from the Case Base, building on the work of Hanney and Keane (Hanney & Keane, 1996). One of the most important aspects in automatic rule generation is the limiting of the number of rules, in order to maintain system functionality (Hanney & Keane, 1997). The most straightforward way of achieving this is to limit the case pairs one accepts for comparison. This can be done by a threshold similairty measure or a threshold number of features in which the cases differ. Within this work, a combination of the two is suggested, as to limit both the effort required for rule generation, as well as the number of feature differences. The threshold similarity is mainly responsible for limiting the search space, while the maximum feature difference ensures that the retrieved rules are applicable. Should two cases differ in more than a certain number of features, the predictive value of an obtained solution would not be sufficient as to provide reliable decision support.

Another way of reducing the number of rules is the elimination of duplicates. In doing so, the confidence rating of the rules is added, since the more often a similar rule is observed within the Case Base, the higher the likelihood of it being correct.

Definition 16 – Rule Generation Algorithm

Input:

- *Case Base CB of m previous cases with pairwise Overall Similarity Measure $S_{a, b}$ and Adaptation Knowledge Requirement tuple $AKR_{a, b} = (f_d , v_d , f_c)$*
- *Rule Base RB populated with domain knowledge rules, which are characterized by a confidence rating of conf = 1*
- *Threshold similarity t_s and threshold number of differnt features t_f*

Output:

- *Enhanced Rule Base RB with Adaptation Rules derived from differences observed within previous cases of the Case Base CB*

Step 1: Rule Generation

Starting with a = 1, For a ≤ m Begin

Starting with b = a + 1, For b ≤ m Begin

If $S_{a, b} > t_s$ and $N (f_d) > t_f$

- *Include all features f_i with $f_{a, i} = f_{b, i}$ into f_c*
- *Include all features f_i with $f_{a, i} \neq f_{b, i}$ into f_d*
- *Set $v_d = (v_{a, d} , v_{b, d})$ such that $v_{a, d} \rightarrow v_{b, d}$ represents the antecedent of the rule*
- *Set $A (C_s) \rightarrow C_s'$ such that the differences in solution features as $(s_{b, i} - s_{a, i})$ represent the consequent of the rule*
- *Set conf = 1 / m*
- *Include Rule with uniquely generated ID in Rule Base*

Else End

End

End

Step 2: Delete Rules conflicting with Domain Knowledge

For all rules j with conf < 1 Start

If $f_{j,d}$ of rule j is equivalent to a known adaptation rule (conf = 1)

Delete Rule j

If $f_{j,d}$ of rule j contains the feature f_i of a rule k with conf = 1 AND $f_{k,d} = f_i$

AND rule $f_{j,c}$ of rule j contains $f_{k,c}$

Delete delete f_i from $f_{j,d}$, as it is irrelevant

End

Step 3: Delete Duplicate Rules

If $f_{j,d} = f_{k,d}$ AND A_j (C_s) and A_k (C_s) are similar, then

- *Set A_j (C_s) as the average of A_j (C_s) and A_k (C_s) for numerical values or the cartesian product for linguistic features*
- *Set $conf_j = conf_j + conf_k$*
- *Delete Rule k*

End

As can be seen in the introduced algorithm, it is possible to use the knowledge already contained within the Case Base to obtain adaptation rules. With the help of this approach to knowledge discovery in the database, feature differences existing between different cases are examined and rules are retrieved on how these differences relate to differences in the case solution.

After generating a set of rules from cases satisfying the similarity and feature difference thresholds, the rule set is further refined. In doing so, the domain knowledge represented in rules is used before similar duplicate rules are merged to achieve a concise representation of the adaptation knowledge in the Rule Base.

With the involvement of domain experts in the rule generation process, the quality of the adaptatin rules is further enhanced and the predictive quality of the matching of the retrieved cases to the current case is improved. The described algorithm describes an automated way of obtaining rules from the Case Base and therefore resloves the issue of the Adaptation Knowledge Bottleneck (Hanney & Keane, 1997). In the course of the following section, the application of the generated rules is introduced in detail, before describing ways of maintaining a high quality Knowledge Base by incorporating the knowledge attained during a given change process.

F.3 Adaptation Rule Application

It is the goal of applying adaptation rules to retrieved similar Engineering Change cases to obtain a high quality estimation of the impact resulting from the change initiative. The first step in doing so is to retrieve cases, which are similar to the current case. This is done by analizing the case description C_d, which is the mapping of the cases on the problem space, and selecting a number of cases in the direct vicinity. These similar previous cases already project an approximate solution C_s into the solution space and therefore provide a first estimation of the impact of the current case. This approximation is further refined with the use of adaptation knowledge. Based on the implicit assumption that the differences occuring between previous cases are representative of the differences between the retrieved cases and the current case, adaptation knowledge in the form of rules is applied and an estimation with a high predictive value is achieved.

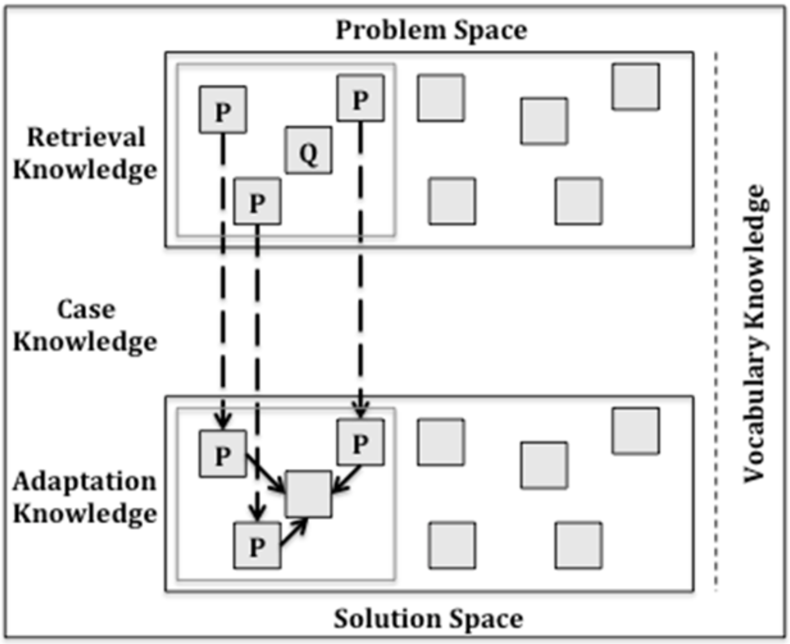

Figure 70 – Solution Process of the System – Based on (Wilke & Bergmann, 1998)

Within the domain of Engineering Change Management, some specific adaptation requirements can not be met by a single rule derived from the Case Base. Therefore, it is necessary to combine rules and to decide on an adaptation path.

Definition 17 – Rule Application Algorithm

Input:

- *New Case* C_p *described with* $C_{p,d} = \{f_{p,1}, \ldots, f_{p,n}, O_{p,n+1}, \ldots, O_{p,j}\}$
- *A similar previous case* C_q

- *Adaptation Knowledge Requirement tuple* $AKR_{p, q} = (f_{pq, d}, v_{pq, d}, f_{pq,}c)$ *with* $v_{pq, d}$ *such that* $v_{q, d} \rightarrow v_{p, d}$ *represents the antecendents required for adaptation*
- *Rule Base RB*

Output:

- *Estimation of the impact of the new case* $C_{p, s}$ *, based on the previous case* C_q *and adaptation rules obtained from the Rule Base of the strategy layer*

Step 1: Find Adaptation Path

If there are rules j , ... , k with $v_{j, d} = ... = v_{k, d} = v_{pq, d}$

Choose rule with highest confidence rating conf

Else For each feature $f_i \in f_d$

Remove feature f_i *from* f_d

Find rules for f_d *satisfying* $v_{j,d} + v_{i, d} = v_{pq, d}$

Choose adaptation path j, i with highest confidence rating $conf_j \times conf_i$

Until Adaptation Paths are all explored

Step 2: Apply Adaptation Routine

Apply $A_j , ... , A_i$ *of the chosen Adaptation Path to the solution* $C_{q, s}$ *of the similar*

previous case C_q *to obtain estimation of impact of new case* $C_{p, s}$

The basis for case adaptation to a current Engineering Change case is a similar previous case. This previously observed instance of a change initiative already provides an approximation for the impact of the current case. The preceding algorithm is based on the assumption that the more specific the adaptation rule is to the current Adaptation Knowledge Requirement, the better is the mapping and transformation of the previous solution. Therefore, it is first attempted to find an adaptation rule exactly matching the feature differences between the two cases.

In case no such rule can be found, the algorithm attempts to construct an adaptation path by subsequently dropping each difference from the original list of differences and searching for a combination of rules fulfilling the entire adaptation requirements. Out of the adaptation paths, the one with the highest aggregate confidence rating is chosen.

During the final step, the chosen adaptation rules are applied to the solution of the retrieved similar case and an estimated impact of the current case is obtained. This solution is then used by the decision maker to evaluate whether the proposed change initiative is further pursued.

At this point, the impact evaluation of an EC case can be undertaken, but the cycle of the system is not concluded, as the knowledge obtained in the course of the Engineering Change initiative needs to be retained for later use.

F.4 Update of the Knowledge Base

In order to continually improve the performance of the Knowledge-Based system, the Case Base is updated upon conclusion of an Engineering Change initiative. The most important aspect is to include the complete description of the case, as well as the observed impacts the change process actually had. Some practitioners warn about retaining every case, as that might lead to the so called swamping problem (Patterson, et al., 1999). However, this appears to be a negligible concern within the domain of Engineering Change Management.

Whenever a new case is introduced to the Case Base, a similarity assessment to the existing cases is done, together with an examination of the feature differences to those cases identified to be similar. On the basis of that, an iteration of the rule generation algorithm is started, which yields the adaptation rules to be retrieved when comparing the new and all similar previous cases.

An additional way of retaining knowledge gathered during an Engineering Change project is by manually implementing the tacit rules learned. It is especially important to include this step in the fixed process model of Engineering Change Management, in order to retain the lessons learned from the experts involved within the process.

Having introduced all the components of the system, namely the Knowledge Base, the Similarity Assessment and the Adaptation Knowledge Retrieval and Application, the following chapter will focus on the architecture of the system and the interplay of the knowledge and the problem solving techniques. Only through a coherent modeling of the information and the different algorithms is it possible to achieve a high performing system, which truly supports the decision making process in the step of Impact Evaluation of the Engineering Change process.

Bibliography

Aamodt, A. & Plaza, E., 1994. Case-based reasoning: foundational issues, methodological variations and system approaches. *Artificial Intelligence Communications,* Issue 7(1), pp. 39-59.

Ahmad, N., Wynn, D. & Clalrkson, P., 2012. Change impact on a product and its redesign process: a tool for knowledge capture and reuse. *Research in Engineering Design.*

Allweyer, T. & Knuppertz, T., 2009. *EDEN–Reifegradmodell Prozessorientierung in Unternehmen.,* s.l.: s.n.

Anand, S., Patterson, D., Hughes, J. & Bell, D., 1998. Discovering case knowledge using data mining. *Proceedings of the Pacific-Asia Conference in Knowledge Discovery and Data Mining.*

Anderl, R. & Trippner, D. (., 2000. *STEP STandard for the Exchange of Product Model Data - Eine Einführung in die Entwicklung, Implementierung und industrielle Nutzung der Normenreihe ISO 10303 (STEP).* Stuttgart: B.G. Teubner.

ANSI/EIA-649, 2011. *National Consensus Standard for Configuration Management.* s.l.:s.n.

Arian, F. & Pheng, L., 2006. *Framework for developing a Knowledge-Based Decision Support System for Management of Variation Orders for Institutional Buildings.* s.l., s.n., pp. 285-293.

Aristotle, 2006. *Metaphysics.* Oxford, UK: Oxford University Press.

Babich, W. A., 1986. *Software configuration management: coordination for team productivity.* New York: Addison-Wesley.

Baccarini, D., 1996. The concept of project complexity - a review. *International Journal of Project Management*, pp. 201-204.

Barber, P., Buxton, I., Stephenson, H. & Ritchey, I., 19. *Design for upgradeability: extending the life of large made to order products at the design stage.* Cambridge, UK, University of Cambridge Institute for Manufacturing, pp. 75-85.

Barret, D., 2002. Change Communication: using strategic employee communication to facilitate major changes. *Corporate Communications: an International Journal*, pp. 219-231.

Batenburg, R., Helms, R. W. & Versendaal, J., 2006. PLM roadmap: stepwise PLM implementation based on the concepts of maturity and alignment.. *International Journal of Product Lifecycle Management,* pp. 1(4), 333-351.

Batenburg, R., H. R. W. & Versendaal, J., 2006. PLM roadmap: stepwise PLM implementation based on the concepts of maturity and alignment.. *International Journal of Product Lifecycle Management, ,* pp. 1(4), 333-351..

Batenburg, R. & Versendaal, J., 2008. Maturity Matters: Performance Determinants of the Procurement Business Function. *ECIS,* pp. pp. 563-574.

Bauer, K., 2004. KPIs-The metrics that drive performance management.. *Information Management.*

Becker, T., 2008. *Prozesse in Produktion und Supply Chain optimieren.* s.l.:Springer.

Belanche, L. & Orozco, J., 2011. Things to Know about a (dis)similarity Measure. In: *KES 2011, Part I.* s.l.:s.n., pp. 100-109.

Benantar, M., 2006. *Access Control Systems.* s.l.:Springer.

Blair, J. C. R. F. &. B. E. A., 2013. *Designing surveys: A guide to decisions and procedures..* s.l.:sage publications.

Bogaerts, S. & Leake, D., 2013. *IUCBRF Indiana University Case-Based Reasoning Framework.* [Online]
Available at: http://www.cs.indiana.edu/~leake/iucbrf/index.html
[Zugriff am 10 08 2014].

Bohn, R., 1995. Measuring and managing techological knowledge. *Sloan Management Review,* Issue 36(1), pp. 61-73.

Booch, G., 1993. *Object-oriented Analysis and Design with Applications.* 2nd Edition Hrsg. s.l.:Addison-Wesley Press.

Borst, P., Akkermansh, H., Pos, A. & Top, J., 1995. The PhysSys ontology for physical systems. In: *The PhysSys.* Amsterdam: Department of Social Science Informatics, University of Amsterdam.

Bourne, L., 2009. *Stakeholder Relationship Management.* s.l.:Gower Publishing.

CERN, 2008. *The Proton Synchrotron.* [Online]
Available at: http://public.web.cern.ch/public/en/Research/PS-en.html
[Zugriff am 15 09 2012].

Chen, C. & Huang, C., 2001. A multiple criteria evaluation of high-tech industries for the science-based industrial park in Taiwan. *Information & Management,* Issue 41, pp. 839-851.

Clark, K. & Fujimoto, T., 1991. *Product development performace: strategy, organisation and management in the world auto industry.* Boston, USA: Harvard Business School Press.

Clarkson, P., Simons, C. & Eckert, C., 2001. Predicting change propagation in complex design. *Proceedings of ASME design engineering conferences.*

CMII, I. o. C. M., 2003. *"ISO 9001 and ISO 10007 Quality Management – Guidance for CM relative to CMII (Rev. B)".*
[Online]

Available at: http://www.configonline.co.za/white%20papers/ISO_10007_Relative_to_CMII.pdf
[Zugriff am 15 07 2012].

CMMI, 2010. *CMMI Product Team, CMMI for Development, Version 1.3 (CMU/SEI-2010-TR-033,* s.l.: Software Management Institute (SEI), Cornegie Mellon University.

Cohen, T., Navathe, S. & Fulton, R., 2000. C-FAR, change favourable representation. *Computer-Aided Design,* Issue 32, pp. 321-338.

Cooper, L., 2006. A Research Agenda to Reduce Risk in New Product Development Through Knowledge Management - A Practitioners Perspective. *Journal of Engieering Technology,* Issue 20(2), pp. 117-140.

Crawford, J., 2007. *Project management maturity model.* s.l.:Boca Raton, FL: Auerbach Publications - Taylor & Francis.

Craw, S., Wiratunga, N. & Rowe, R., 2006. Learning adaptation knowledge to improve case-based reasoning. *Artificial Intelligence,* pp. 1175-1192.

Cross, N., 1989. *Engineering design methods.* Wiley: s.n.

Davenport, T. & Prusak, L., 1998. Working Knowledge. *Harvard Business Review,* Issue 42(2), pp. 18-23.

de Bruin, T., Rosemann, M., Freeze, R. & Kulkarni, U., 2005. *Understanding the main phases of developing a maturity assessment model..* s.l., s.n.

Denger, A. & Unzeitig, W., 2012. *Future product lifecycle management (plm) - a consideration of Informal communication as a key enabler for future product development.* s.l., s.n., pp. 435-444.

Diaz-Agudo, B. & Gonzalez-Calero, P., 2007. An Ontological Approach to develop Knowledge Intensive CBR Systems. In: *Ontologies - A Handbook of Principles, Concepts and Applications in Information Systems.* New York, USA: Springer, pp. 173-213.

DiPrima, M., 1982. Engineering change control and implementation considerations. *Product Inventory Management*, Issue 23, pp. 81-87.

Doppler, K., 2011. *Unternehmenswandel gegen Widerstände, Change Management mit den Menschen.* Frankfurt am Main: Campus Verlag.

Dutta, D., W. J. P. & Wolowicz, J., 2005. An introduction to product lifecycle management (PLM). *ISPE.*

Dvir, D. L. S. S. A. &. T. A., 1998. . In search of project classification: a non-universal approach to project success factors. *research policy,* pp. 915-935.

Eckerson, W., 2006. *Performance Dashboards: Measuring, monitoring and managing your business.* Hoboken N.J.: John Wiley.

Eckert, C., Clarkson, P. & Earl, C., 2004. Change and customisation in complex engineering domains. *Res Eng Des 15(1)*, pp. 1-21.

Eckert, C., Wyatt, D. & Clarkson, P., 2009. *The elusive act of synthesis: creativity in conceptual design of complex engineering products.* Berkley, USA, s.n., pp. 265-274.

Elkington, J., 1997. Cannibals With Forks; The Triple Bottom Line of the 21st Century. *New Society*.

Fischermanns, G., 2009. *Praxishandbuch Prozessmanagement.* s.l.:Verlag Dr. Götz Schmidt, Gießen.

Fowler, A., 1993. Models and applications of configuration management. *Omega,* pp. 21(4), 425-431..

Fraser, P., Moultrie, J. & Gregory, M., 2002. *The use of maturity models / grids as a tool in assessing product development capability.* s.l., Engineering Management Conference, 2002. IEMC'02 - 2002 IEEE International.

Fricke, E., Gebhard, B., Neglel, H. & Ingenbergs, E., 2000. Coping with changes: causes, findings and strategies. *Systems Engineering 3(4)*, pp. 169-179.

GAIA, 2013. *Group of Artificial Intelligence Applications.* [Online]
Available at: http://gaia.fdi.ucm.es/about
[Zugriff am 2013].

GEIA, G. E. a. I. T. A., 2002. *Systems Engineering Capability Model (EIA-731.1).* s.l.:Electronic Industries Alliance, EIA.

Gensereth, M. & Nilsson, N., 1987. *Logical Foundations of Artificial Intelligence.* Los Altos, CA, USA: Morgan Kaufmann.

Gmelin, H. & Seuring, S., 2014. Achieving sustainable new product development by integrating product life-cycle management capabilities. *International Journal of Production Economics,* pp. 166-175.

Gomez Perez, A. & Benjamins, V., 1999. *Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods.* Stockholm, Sweden, s.n., pp. 1-15.

Gomez Perez, A., Fernandez Lopez, M. & Corcho, O., 2003. *Ontological Engineering.* London, UK: s.n.

Gonzalez, P., 2002. *A Guide to Configuration Management for Intelligent Transportation Systems,* s.l.: s.n.

Gruber, T., 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition,* Issue 5(2), pp. 199-220.

Guarino, N. & Giaretta, P., 1995. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In: *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing.* Amsterdam, NL: IOS Press.

Guarino, N., Oberle, D. & en Staab, S., 2009. What Is an Ontology?. In: *Handbook on Ontologies.* Heidelberg: Springer, pp. 1-17.

Guess, V. C., 2005. *Change Management White Paper,* s.l.: Institute of Configuration Management.

Guess, V. C., 2013. *CMII: The Path to Integrated Process Excellence.* s.l.:CMII Research Institute.

Gupta, P. & Rao, D., 2011. Best Practices to Achieve CMMI Level 2 Configuration Management Process Area through VSS tool. *Int. Journal of Computer Technology And Applications.*

Hall, M. et al., 2009. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter,* pp. 10-18.

Hamraz, B., 2013. *Engineering Change Modelling using a Function-Behaviour-Structure Scheme,* Cambridge, UK: s.n.

Hanney, K. & Keane, M., 1996. Learning adaptation rules from a case-base. In: *Advances in Case-Based Reasoning - Third European Workshop EWCBR-96.* Lausannne, Switzerland: Springer, pp. 179-192.

Hanney, K. & Keane, M., 1997. The Adaptation Knowledge Bottleneck: How to Unblock it By Learning from Cases.

Harris, A., 2009. *A knowledge based framework to support product development,* Wolverhampton, UK: s.n.

Hass, A. M. J., 2003. *Configuration Management principles and practice.* s.l.:Addison-Wesley, Boston, MA.

Heinecke, A. M., 2012. *Mensch-Computer-Interaktion.* s.l.:Springer.

Huang, G. & Mak, K., 1998. Computer aids for engineering change control. *Journal of Materials Processing Technology*, Issue 76, pp. 187-191.

Huang, G. & Mak, K., kein Datum Current practices of engineering change management in UK manufacturing industries. *International Journal of Operations and Production Management*, Issue 19(1), pp. 21-37.

Humphrey, W., 1987. *Characterizing the Software Process: A Maturity Framework -,* s.l.: Software Engineering Institute, CMU/SEI-87-TR-11, ADA182895 .

IAEA, I. A. E. A., 2010. *Application of Configuration Management in Nuclear Power Plants.* s.l.:Safety Reports Series No. 65.

IAQG, I. A. Q. G., 2011. *Configuration management guidelines.* [Online]
Available at: http://www.sae.org/iaqg

IEEE-STD-610, 1990. *IEEE Standard Glossary of Software Engineering Terminology,* s.l.: E Iee.

Institute of Electrical and Electronics Engineers, 1990. *IEEE 610.12-1990 Standard glossary of software engineering terminology ,* New York, USA: s.n.

International Organisation for Standardisation, 2003. *Quality management systems -- Guidelines for configuration management,* s.l.: s.n.

International Organisation for Standardisation, 2008. *Quality management systems -- Requirements,* s.l.: s.n.

International Organisation for Standardization, 2006. *ISO 11442:2006: Technical product documentation -- Document management,* s.l.: s.n.

ISO/IEC 12207, 2008. *Systems and Software Engineering – Software Life cycle Processes.* s.l.:Software and Systems Engineering Standards Committee, International Standard Organization (ISO).

ISO/IEC TS 15504, 2011. *Information technology -- Process assessment.* s.l.:ISO.

ISO/IS 10007, 2003. *Quality Management Systems - Guidelines for Configuration Management.* s.l.:ISO.

ISO/IS, 2000. *10303-224.2 Product data representation and exchange: Application protocol: Mechanical product definition for process planning using machining features.* s.l.:ISO.

ISO/IS, 2001. *10303-203: Product data representation and exchange: Application protocol: Configuration controlled design.* s.l.:ISO.

ISO/IS, 2001. *10303-212: Product data representation and exchange; Application protocol: Electrotechnical design and installation.* s.l.:ISO.

ISO/IS, 2002. *10303-212: Product data representation and exchange; Application protocol: Technical data packaging core information and exchange.* s.l.:ISO.

ISO/IS, 2003. *10303-212: Product data representation and exchange; Application protocol: Core data for automotive mechanical design processes.* s.l.:ISO.

ISO/IS, 2003. *Product data representation and exchange: Application prodocol: Systems engineering and data representation.* s.l.:ISO.

ISO/IS, 2005. *10303-239 Product data representation and exchange: Application protocol: Product life cycle support.* s.l.:ISO.

ISO/IS, 2005. *10303-240: Product data representation and exchange: Application protocol: Plans for machined Products.* s.l.:ISO.

Jarratt, T., Eckert, C. & Caldwell, N., 2010. *Engineering Change: an overview and perspective on the literature,* London: s.n.

Jarratt, T., Eckert, C. & Clarkson, P., 2004. Engineering Change. In: *Design process improvement.* New York: Springer.

Joshi, N., 2007. *Methodologies for improving product development phases through PLM.* University of Michigan, USA: s.n.

Kärkkäinen, H., Pels, H. & Silventoinen, A., 2012. *Defining the customer dimension of PLM maturity..* s.l., In: Rivest, L., Bouras, A., Louhichi, B. (eds.) PLM 2012. IFIP AICT, vol. 388, pp. 623–634. Springer, Heidelberg (2012).

Kasvi, J., Vartianen, M. & Hailikari, M., 2003. Managing Knowledge and Knowledge Competencies in Projects and Project Organizations. *International Journal of Project Management,* Issue 21, pp. 571-582.

Kepinski, M. et al., 2013. *TREC: TRACEABILITY OF RADIOACTIVE EQUIPMENT AT CERN.* Shanghai, IPAC, THPEA042.

Kidd, M. & Thompson, G., 2000. Engineering design change management. *Integrated Manufacturing System*, Issue 11(1), pp. 74-77.

Klingebiel, N., 2013. *Performance Measurement: Grundlagen—Ansätze—Fallstudien.* s.l.:Springer-Verlag.

Kocar, V. & Akgunduz, A., 2010. ADVICE: a virtual environment for engineering change management. *Computers in Industry,* Issue 61(1), pp. 15-18.

Kohn, A., Lindemann, U. & Maurer, M., 2013. *Knowledge base for supporting the handling of product models in engineering design.* Seoul, Korea, s.n.

Körfer, R., 2006. Merger Integration - die ultimative Herausforderung des Change Management. In: *Strategisches Management von Mergers & Acquisitions.* Wiesbaden: Betriebswissenschaftlicher Verlag Dr. Th. Gabler, pp. 279-295.

Kulkarni, U. & Freeze, R., 2004. *Development and Validation of a Knowledge Management Capability Assessment Model.* s.l., Twenty-Fifth International Conference on Information Systems .

Kuvaja, P. & Bicego, A., 1994. Bootstrap – A European Assessment Methodology. *Software Quality Journal,* pp. (3), 117–127 .

Lampson, B. W., 1971. *Protection.* s.l.:Proceedings of the 5th Princeton Conference on Information Sciences and Systems.

Leake, D., Kinley, A. & Wilson, D., 1995. Learning to Improve Case Adaptation by Introspective Reasoning and CBR. In: *Case-based Reasoning Research and Development: Lecture Notes in Artificial Intellicence 1010.* s.l.:Springer Verlag, pp. 229-240.

Leake, D., Kinley, A. & Wilson, D., 1996. Acquiring Case Adaptation Knowledge: A Hybrid Approach. *AAAI/IAAI, Volume 1,* pp. 684-689.

Lee, H., Ahn, H., Kim, J. & Park, S., 2006. Capturing and reusing knowledge in engineering change management: A case of automobile development. *Inf Syst Front,* pp. 375-394.

Lee, M., 2003. A study of automatic learning model of adaptation knowledge for case base reasoning. *Information Sciences,* Issue 155(1), pp. 61-78.

Lehner, F., 2009. *Wissensmanagement: Grundlagen, Methoden und technische Unterstützung, 3rd ed..* München, Wien: Hanser.

Lenat, D. & Guha, R., 1990. *Building large knowledge-based systems: Representations and Interface in the Cyc Project.* Boston, MA,USA: Addison-Wesley.

Lindkvist, C., Stasis, A. & Whyte, J., 2013. *Configuration management in complex engineering projects.* s.l., Elsevier, B.V., pp. 173-176.

Liu, P., Chen, W. & Tsai, C., kein Datum An Empirical Study on the Correlation Between the Knowledge Management Method and New Product Development Strategy on Product Performance in Taiwan's Industries. *Technovation,* Issue 25(6), pp. 637-644.

Liu, S., Duffy, A., Whitfield, R. & Boyle, I., 2010. Integration of decision support systems to improve decision support performance. *Knowledge Information Systems,* Issue 22, pp. 261-286.

Lyon, D. D., 1999. *Practical CM, Best Configuration Management Practices.* s.l.:Butterworth-Heinemann.

Lyon, D. D., 2000. *Practical CM: Best Configuration Management Practices.* s.l.:Butterworth-Heinemann..

Maier, R., 2004. *Knowledge Management Systems, Information and communication technologies for knowledge management.* Berlin, Heidelberg: Springer.

Man, T.-J., 2007. A framework for the comparison of Maturity Models for Project-based Management. *Utrecht University.*

Maropoulos, P. & Ceglarek, D., 2010. Design verification and validation in product lifecycle. *CIRP Annals - Manufacturing Tecnology*, Issue 59, pp. 740-759.

Martin, J., 1990. *Information Engineering. A triology, Book II,* Englewood Cliffs N.J.: Prentice Hall.

Mehta, C., 2010. *Knowledge-based Methods for Evaluation of Engineering Changes,* s.l.: s.n.

Melnik, S. & Decker, S., 2000. *A layered approach to information modeling and interoperability.* Lisbon, Portugal, s.n.

Mettler, T., 2009. *A Design Sciene Research Perspective on Maturity Models in Information Systems.,* s.l.: Institute of Information Management, St. Gallen.

Mettler, T., Rohner, P. & Winter, R., 2010. Towards a classification of maturity models in information systems. *In Management of the interconnected world - Physica-Verlag HD.,* pp. 333-340.

MIL-HDBK-61 , 1997. *Military Handbook. Configuration Management Guidance..* s.l.:Department of Defense.

MIL-STD-3046, 2013. *Interim Standard for Configuration Management.* s.l.:Department of Defense (DOD) .

Mitra, R. & Basak, J., 2005. Methods of Case Adaptation: A Survey. *International Journal of Intelligent Systems,* Issue 20, pp. 627-645.

Moen, R. &. N. C., 2006. Evolution of the PDCA cycle. In: s.l.:s.n.

Monahan, R., 1995. *Engineering documentation control practices and procedures.* New York: Marcel Dekker Inc,.

Mottier, M., 1999. *Quality Assurance Policy,* Geneva Switzerland: CERN.

Murray, M., 2012. *The Ionising Radiations Regulations 1999 (IRR'99): The Guidance Booklet,* s.l.: Society of Radiographers (www.sor.org).

Nahar, J., Imam, T., Tickle, K. S. & Chen, Y.-P. P., 2013. Association rule mining to detect factors which contribute to heart disease in males and females. *Expert Systems with Applications,* Issue 40, pp. 1086-1093.

Nielson, J., 1993. *Usability Engineering.* s.l.:Academic Press, San Diego.

Niknam M, Ovtcharova J, Bierbrauer S & R, T., 2013. *A cross-industry assessment of Configuration Management Maturity,* s.l.: P3-Group Publishing.

Niknam, M., Huang, E. & Ovtcharova, J., 2014. *Towards a Decision Support System for Engineering Change Management.* Dubrovnik, Croatia, s.n.

Niknam, M. & Ovtcharova, J., 2013. Towards Higher Configuration Management Maturity. In: *Product Lifecycle Management for Society.* Berlin: Springer, pp. 396-404.

Noran, O., 2009. A Decision Support System for Enterprise Engineering. In: *Information Systems Development: Challenges in Practice, Theory and Education.* s.l.:Springer-Verlag, pp. 533-546.

Ollinger, G. & Stahovich, T., 2001. *"RedesignIT" - A constraint-based tool for managing design changes.* Pittsburgh, USA, s.n.

Ovtcharova, J., 2012. *Lecture in Product Lifecyle Management.* s.l.:Karlsruhe Institute of Technology.

Ovtcharova, J., Weigt, M. & Seidel, M., 2005. *Virtual Engineering.* s.l., Proceedings of 7th Magdeburg Days of Mechanical Engineering in Virtual Development of Products and Processes pp 9-14.

Owen, J., 1993. *STEP: An introduction.* s.l.:Information Geometers.

P.C. Patton, B. J., 2007. *Design for Trustworthy Software.* s.l.:Pearson Education Inc..

Parmenter, D., 2007. *Key Performance Indicators: Developing, Implementing and Using Winning KPIs, 2nd. ed.*. Hoboken N.J.: John Wiley.

Patterson, D., Anand, S., Dubitzky, W. & Hughes, J., 1999. Towards Automated Case Knowledge Discovery in the M2 Case-Based Reasoning System. *Knowledge and Information Systems,* Issue 1, pp. 61-82.

Patterson, D., Dubitzky, W., Anand, S. & Hughes, J., 1998. On the Automation of Case Base Development from Large Databases. *AAAI Technical Report,* Issue WS-98-15, pp. 126-130.

Pikosz, P. & Malmqvist, J., 1998. A comparative study of engineering change management in three Swedish engineering companies. *Proceedings of ASME design engineering technical conferences.*

Plaisant, B. S. a. C., 2010. *Designing the User Interface.* s.l.:Pearson Education Inc..

PLM Interest Group, 2014. *PLM Maturity Reference Model.* [Online]
Available at: http://www.plmig.com/
[Zugriff am 14 03 2014].

PMI, P. M. I., 2010. *A Guide to Project Management Body of Knowledge.* s.l.:Project Management Institute.

Policastro, C., Carvalho, A. & Delbem, A., 2006. Automatic knowledge learning and case adaptation with a hybrid committee approach. *Journal of Applied Logic,* Issue 4(1), pp. 26-38.

Power, D., 2000. *Decision Support System Hyperbook.* [Online]
Available at: http://dssresources.com/subscriber/password/dssbookhypertext.
[Zugriff am 18 06 2014].

Power, D. J., 2008. Decision Support Systems: A Historical Overview. In: *Handbook on Decision Support Systems 1.* Heidelberg: Springer-Verlag, pp. 121-140.

Prentazas, J. & Hatzilygeroudis, I., 2007. Categorizing approaches combining rule-based and case-based reasoning. *Expert Systems,* Issue 24(2), pp. 97-122.

RDF, 1999. *Resource Description Framework.* [Online]
Available at: http://www.w3.org/TR/rec-edf-syntax/
[Zugriff am 26 07 2014].

Recio-Garcia, J., Diaz-Agudo, B., Gonzalez-Calero, P. & Sanchez-Ruiz-Granados, A., 2007. Ontology based CBR with jCOLIBRI. *Applications and Innovations in Intelligent Systems,* Issue 14, pp. 149-162.

Recio-Garcia, J., Gonzalez-Calero, P. & Diaz-Agudo, B., 2014. jCOLIBRI2: A framework for building Case-based reasoning systems. *Science of Computer Programming,* Issue 79, pp. 126-145.

Reidelbach, M., 1991. Engineering change management for long -lead-time produciton environments. *Production Inventory Management Journal*, Issue 32, pp. 84-88.

Resnik, P., 1999. Semantic Similarity in a Taxonomy: An Information Based Measure and its Application Natural to Problems Language. *Journal of Artificial Intelligence Research,* 7, Issue 11, pp. 95-130.

Riviere, A., DaCunha, C. & Tollenaere, M., 2002. *Performances in Engineering Changes Management.* Clemont-ferrand, France, s.n.

Robbins, S., De Cenzo, D. & Coulter, M., 2013. *Fundamentals of management.* Boston (Mass.): Pearson.

Röglinger, M., Pöppelbuß, J. & & Becker, J., 2012. Maturity models in business process management. *Business Process Management Journal,* pp. 18(2), 328-346..

Rohm, H., 2008. *Is There Any Strategy In your Strategic Plan?,* Cary, North Carolina: Balanced Scrorecard Institute.

Rouse, W. & Sage, A., 2011. *Handbook of Systems Engineering and Management.* s.l.:Jon Wiley & Sons.

Ryu, K. S., Park, J. S. & Park, J. H., 2006. A data quality management maturity model.. *ETRI journal,* pp. 28(2), 191-204..

Saaksvuori,, A. & Immonen, A., 2008. *Product lifecycle management..* s.l.:Springer Science & Business Media..

Saaty, T., 1980. *The analytic hierarchy process.* New York, USA: McGraw-Hill.

Saaty, T. & Vargas, L., 2012. The Seven Pillars of the Analytic Hierarchy Process. In: *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process.* New York: Springer-Verlag, pp. 23-40.

Saco, R. M., 2008. Maturity Models. *Industrial Management,* p. 50(4).

Sajja, P. & Akerkar, R., 2010. *Advanced Knowledge Based Systems: Model, Application & Research.* s.l.:s.n.

Salton, G. & McGill, M., 1983. *Introduction to Modern Information Retrieval.* New York, USA: McGraw-Hill.

Samaras, T. T., Humm, M. J. & Illustrator-Robel, C., 1988. *Configuration management deskbook..* s.l.:Advanced Applications Consultants, Incorporated..

Sanchez, D., Cavero, J. & Martinez, E., 2007. The Road towards Ontologies. In: *Ontologies - A Handbook of Concepts and Applications in Information Systems.* s.l.:Springer, pp. 3-20.

Savino, M., Mazza, A. & Ouzrout, Y., 2012. *PLM Maturity Model: A Multi-Criteria Assessment in Southern Italy Companies.* s.l., International J. of Opers. and Quant. Management.

Scheper, W., 2002. *Business IT Alignment: Solution for the productivity paradox (In Dutch),* s.l.: Deloitte & Touche.

Schneiderman, B., 1996. *The eyes have it: A task by data type taxonomy for information visualizations.* s.l.:Department of Computer Science, Human-Computer Interaction Laboratory, and Institute for Systems Research University of Maryland.

Scholl, T., 2012. *Titel der Veröffentlichung.* München: s.n.

Scupin, R., 1997. The KJ method: A technique for analyzing data derived from Japanese ethnology. *Human organization,* pp. 56(2), 233-237.

Shaofen, L., Duffy, A. H., Whitfield, R. I. & Boyle, I. M., 2010. Integration of decision support systems to improve decision support performance. *Knowledge Information Systems,* Issue 22, pp. 261-286.

Siemens Product Lifecycle Management Software Inc., 2010. *Change Management,* Munich: s.n.

Smyth, B. & Keane, M., 1995. *Remembering to Forget: A Competence-Preserving Deletion Policy in Case-Based Systems.* s.l., s.n.

Softic, S. et al., 2014. *Ontology based tool for task tracking and decision support in an automotive collaborative environment.* Dubrovnik, Croatia, s.n., pp. 1895-1904.

Softic, S. et al., 2014. *Ontology based tool for Task Tracking and Decision Support in an Automotive Collaborative Working Environment.* Dubrovnik, Croatia, Design 2014, pp. 1895-1904.

Software Engineering Institute (SEI) , 2014. *CMMI.* [Online]
Available at: http://www.sei.cmu.edu/
[Zugriff am 20 03 2014].

Sosa, M., Eppinger, S. & Rowles, C., 2007. A network approach to define modularity of components in complex products. *Journal of Mechanical Design,* Issue 129, pp. 1118-1129.

Stahl, A. & Roth-Berghofer, T., 2008. Rapid prototyping of cbr applications with the open source tool mycbr. In: *Advances in Case-Based Reasoning, 9th European Conference, ECCBR 2008.* Trier, Germany: Springer, pp. 615-629.

Stark, J., 2005. *Product Lifecycle Management: 21st Century Paradigm for Product Realization.* s.l.:Springer, Berlin.

Strazdina, R. & Kirikova, M., 2011. Change Management for Fractal Enterprises. In: *Information Systems Development.* s.l.:s.n., pp. 735-745.

Suh, N., 1990. *The principles of design.* New York: Oxford University Press.

Terwiesch, C. & Loch, C., 1999. Managing the process of engineering change orders: the case of the climate control System in automobile development. *Journal of Product Innovation Management,* Issue 16, pp. 160-172.

Terwiesch, C. & Loch, C., 1999. Managing the process of engineering change orders: the case of the climate control system in the automobile development. *Journal of Product Innovation Management,* pp. 160-172.

The W. Edwards Deming Institute®, 2016. *The PDSA Cycle.* [Online]
Available at: https://www.deming.org/theman/theories/pdsacycle

Thompson, S. M., 1997. Configuration Management - Keeping it all together. *BT technology journal,* pp. 15(3), 48-60..

Turban, E., 1995. *Decision support and expert systems: management support systems.* Englewood Cliffs, N.J.: Prentice Hall.

Ulrich, H., 2001. Unternehmungspolitik. In: *Gesammelte Schriften Band 2.* Bern: s.n.

Ulrich, K., 1995. The role of product archtecture in the manufacturing firm. *Res Policy,* Issue 24, pp. 419-440.

Uschold, M., 2004. Ontologies and Semantics for Seamless Connectivity. *SIGMOD Record,* Issue 33(4), pp. 58-64.

Uschold, M., King, M., Moralee, S. & Zorgios, Y., 1998. The Enterprise Ontology. *The Knowledge Engineering Review,* Issue 13.

Wang, L. & Kourouklis, A., 2013. Knowledge Management for Innovation and Product Development in Supply Chains. In: *IC3K2012, CCIS 415.* s.l.:s.n., pp. 350-376.

Wang, Z. & Wan, Y., 2013. *Research on Engineering Change Knowledge Representation and Retrieval Technology Based on Ontology.* Brunel University, Uxbridge, UK, s.n.

Watts, F. B., 2008. *Engineering Documentation Control Handbook.* s.l.:William Andrew Inc..

Watts, F. B., 2009. *Configuration Management Metrics.* s.l.:Elsevier Inc..

Wendler, R., 2012. The maturity of maturity model research: A systematic mapping study. *Information and software technology,* pp. 54(12), 1317-1339.

Wilke, W. & Bergmann, R., 1998. Techniques and Knowledge used for Adaptation during Case-Based Problem Solving. In: *Tasks and Methods in Applied Artificial Intelligence.* Berlin: Springer, pp. 497-506.

Wilke, W., Vollrath, I., Althoff, K. & Bergmann, R., 1997. *A framework for learning adaptation knowledge based on knowledge light approaches.* s.l., s.n., pp. 235-242.

Wright, I., 1997. A review of research into engineering change management: implications for product design. *Des. Stud.,* Issue 18, pp. 33-42.

Wynn, D., Eckert, C. & Clarkson, P., 2007. *Modelling iteration in engineering design.* Paris, France, s.n., pp. 691-694.

Zhang, H. et al., 2013. *PLM Components Selection Based on a Maturity Assessment and AHP Methodology.* s.l., Bernard, A., Rivest, L., Dutta, D. (eds.) PLM 2013. IFIP AICT, vol. 409, pp. 439–448. Springer, Heidelberg (2013).

Zhou, F. G., Zhang, F. & Yang, B. R., 2009. General Structural Model and Application of Intelligent Decision Support System Based on Knowledge Discovery. *CCIS*, Issue 34, pp. 218-225.