

Konfigurationsmanagement von Modellen

Prof. Dr. Werner Esswein
Dipl.-Wirtsch.-Inf. Steffen Greiffenberg
Dipl.-Wirtsch.-Inf. Christian Kluge
Technische Universität Dresden

Lehrstuhl für Wirtschaftsinformatik, insb. Systementwicklung

Abstract: Die Entwicklung von Methoden zum Konfigurationsmanagement wird als einer der zentralen Erfolgsfaktoren im Software Engineering angesehen (vgl. [Estu00], S. 1). Die Integration entsprechender Verfahren in den Entwicklungsprozess ermöglichte die Erstellung umfangreicher, komplexer Systeme unter Einhaltung der erforderlichen Qualitätsstandards. Die Qualität eines Softwareprodukts wird jedoch nicht allein durch dessen Realisierung, sondern insb. auch durch die frühen Phasen des Designs geprägt (vgl. [PoB193], S. 13).

Gegenstand dieses Entwicklungsabschnitts ist neben der Analyse und Dokumentation der Anforderungen zumeist auch die Erstellung von Modellen über das zu implementierende System. Sie dienen beispielsweise der Darstellung seiner Architektur oder der Zuordnung der Realisierungsarbeiten zu den beteiligten Entwicklern. Mit zunehmendem Umfang der zu realisierenden Systeme steigt aber auch die Komplexität zugehöriger Modelle. Die Einführung von Verfahren zum Konfigurationsmanagement in Modellen ermöglicht ihre verbesserte Beherrschung unter Wahrung der notwendigen Qualitätsstandards. Anliegen dieses Beitrages ist es, für den Bereich der Informationsmodelle geeignete Strukturen und Verfahren zu entwickeln. Im Ergebnis steht aufgrund der Anforderungen relevanter Normen und vorliegender Erfahrungsberichte aus der Softwareentwicklung eine Konzeption zum Konfigurationsmanagement in Modellen, die sich aufgrund der Abstraktion von einem konkreten Meta-Modell universell einsetzen lässt.

0 Einleitung

Die Transformation von implizitem in mitteilbares, öffentliches Wissen stellt nach den Ausführungen von DAHME und RAEITHEL (vgl. [DaRa97], S. 10) unter anderem eine wichtige Aufgabe gegenständlicher Modelle dar. Kenntnisse und Erfahrungen sind jedoch nicht unveränderlich, so dass Modelle Änderungen im Zeitablauf unterliegen. Letztere lassen sich anhand der Modellbildungsdimensionen beschreiben, die von DRESBACH mit dem Frageschema: „... Modell wovon, für wen, wann und wozu.“ ([Dres99], S. 79) zusammengefasst wurden. Die Aufzeichnung der Modifikationen und ihrer Ursachen ermöglicht eine rückwirkende Analyse derselben. Sie kann somit auch Informationen für Entscheidungen in zukünftigen Entwicklungen liefern.

In gleichem Maße eröffnet der zunehmende Einsatz von Referenzmodellen insb. im Bereich der Wirtschaftsinformatik (vgl. [Schü98], S. VII und 1) zusätzliche Anwendungspotentiale für die Aufzeichnung von Modellversionen. Mit der Ableitung individueller Modelle durch Build- bzw. Runtime-Operatoren werden Varianten des Referenzmodells gebildet (vgl. [Schü98], S. 244ff). Ihre Speicherung kann, in Verbindung mit den ebenfalls verwalteten Anpassungsprämissen, die zukünftige Entwicklung von Individualmodellen für vergleichbare Situationen erheblich beschleunigen. Andererseits erlaubt sie aber auch die Analyse von Gemeinsamkeiten der Varianten und unterstützt damit die Entwicklung branchenbezogener Referenzmodelle.

Die dargestellten Potentiale einer Änderungsverwaltung bzw. -kontrolle, sowie die daraus erwachsenden Möglichkeiten zur Qualitätsverbesserung in der Modellerstellung bilden die Motivation für die Entwicklung eines Konzepts zur Versionierung von Modellen im vorliegenden Beitrag. Hierzu werden im Folgenden zunächst grundlegende Begriffe analysiert und ihre Bedeutung im vorliegenden Kontext bestimmt. Im nachfolgenden Teil erfolgt die Untersuchung der Anforderungen an eine Versionierung im Allgemeinen und in Bezug auf Modelle, was die Forderung nach einer entsprechenden Begriffsbestimmung untermauert. Zum Abschluss der Ausführungen erfolgt anhand der Anforderungen eine allgemeine Spezifikation eines Systems zur Versionierung von Modellen.

1 Begriffsbestimmung

1.1 Modellbegriff

Grundlage der Intension des Modellbegriffs bilden in diesem Beitrag die allgemeinen Modellmerkmale nach STACHOWIAK (vgl. [Stac73], S. 131ff), die durch erkenntnistheoretische Betrachtungen ergänzt werden. Aufgrund der Reichweite des damit aufgespannten Begriffsumfangs erfolgt abschließend eine Beschränkung auf den für den Beitrag relevanten Bereich der Informationsmodelle.

Gemäß den Ausführungen von STACHOWIAK, die übereinstimmend auch in der neueren Literatur zur Analyse des Modellbegriffs eingesetzt werden (vgl. auch [Schü98], S. 41f; [Moli84], S. 28ff und [Herr91], S. 109), sind Modelle durch das Abbildungsmerkmal, das Verkürzungsmerkmal und das pragmatische Merkmal gekennzeichnet (vgl. [Stac73], S. 131ff). Sie stellen stets Repräsentationen natürlicher oder künstlicher Originale dar, wobei lediglich die relevanten Aspekte betrachtet werden (Prinzip der Abstraktion, vgl. auch [Stic+97], S. 2). Entsprechend dem pragmatischen Merkmal besitzen sie einen Subjektbezug, denn sie sind „... ihren Originalen nicht per se zugeordnet. Sie erfüllen ihre Ersetzungsfunktion a) für bestimmte ... Subjekte, b) innerhalb bestimmter Zeitintervalle und c) unter Einschränkung auf bestimmte gedankliche oder tatsächliche Operationen.“ ([Stac73], S. 132f).

Für die Bestimmung der Modellintension aus erkenntnistheoretischer Sicht werden in der Literatur zumeist der abbildungsorientierte und der konstruktivistische Modellgedanke herangezogen. Bei Annahme eines abbildungsorientierten Verständnisses ist ein Modell als „... eine vereinfachende und abstrahierende Darstellung eines Realitätsausschnitts ...“ ([Schü98], S. 52) zu interpretieren, „... anhand dessen die wichtigsten Eigenschaften eines Originals erkannt, verstanden und analysiert werden können.“ ([Stic+97], S. 449). Im Gegensatz dazu definiert sich bei Zugrundelegung des Konstruktivismus ein Modell nicht mehr über seine Abbildungsrelation zum Original, sondern als „... das Ergebnis einer Konstruktion eines Modellierers, der für Modellnutzer eine Repräsentation eines Originals zu einer Zeit als relevant mit Hilfe einer Sprache deklariert ...“ ([Schü98], S. 59).

Gegen die abbildungsorientierte Auffassung spricht deren Annahme einer, im Idealfall sogar isomorphen, Abbildungsrelation zwischen Modell und Original. Die dafür notwendige Voraussetzung der objektiven Wahrnehmung der Realität muss in Zweifel gezogen werden (vgl. [Herr91], S. 116ff). Im Falle einer objektiv wahrnehmbaren Realität würden Probleme bereits ihre Lösung beinhalten, die durch geeignete Umformoperationen extrahiert werden kann (vgl. [Dres99], S. 78). Dies ist aber offensichtlich in der heutigen Welt nicht der Fall, so dass nicht von einer objektiv wahrnehmbaren Realität ausgegangen werden kann. Den zweiten Kritikpunkt der Abbildungsorientierung bildet der passiv-rezeptive Modellierungsprozess, nach dessen Verständnis ein Modell durch Beobachtung entsteht. Dies ist mit den oben dargelegten Einschränkungen bez. der Original-Modell-Abbildungsrelation zwar für Istmodelle, allerdings nicht für die Erstellung von Sollmodellen möglich (vgl. [Schü98], S. 58f). Vielmehr konstruiert der Modellierer bei Bildung letzterer eine innere Vorstellung (internes Modell), die durch geeignete Kommunikationssysteme externalisiert wird. Aufgrund dieser Unzulänglichkeiten wird für die weitere Argumentation das konstruktivistische Modellverständnis zugrunde gelegt, bei dem insb. die Bedeutung der Konstruktionsleistung des Modellerstellers hervorzuheben ist.

Durch diese Modellintension wurde ein sehr weiter Begriffsraum aufgespannt, wie er auch den Ausführungen von STACHOWIAK (vgl. [Stac73]), JÄGER (vgl. [Jäge82], S. 144) und MOLIÈRE (vgl. [Moli84], S. 51) entnommen werden kann. Die Entwicklung eines Versionierungskonzepts unter Einsatz des allgemeinen Begriffs erscheint zwar ob der möglichen Generik und Universalität in der Anwendung sinnvoll, ist jedoch insb. im Hinblick auf die Heterogenität der erfassten Modelltypen nicht realisierbar. Aufgrund der Positionierung des Beitrages im Bereich der Wirtschaftsinformatik wird das Begriffsfeld auf den Bereich der Informationsmodelle (IM) beschränkt. Sie können als deren klassischer Modelltyp verstanden werden (vgl. [Schü98], S. 63). Unter einem IM wird hier das „... immaterielle Abbild des betrieblichen Objektsystems aus Sicht der in diesem verarbeiteten Informationen für Zwecke des Informationssystem- und des Organisationsgestalters.“ verstanden (vgl. [Beck+95], S. 435).

1.2 Versionierung und Konfigurationsmanagement

Eine Version stellt allgemein einen Zustand bzw. Aspekt eines Objekts dar (vgl. auch [West91], S. 51; [CoWe96], S. 8; [Rose92], S. 9; [Habe93], S. 209 und [Zell97], S. 9), wobei von der Struktur des versionierten Objekts abstrahiert wird. Eine Anwendung dieser Definition auf Modelle und damit die alleinige Aufzeichnung verschiedener Entwicklungsstände ist unzureichend für die hier verfolgten Zwecke. Um Veränderungen am Modell im Zeitablauf rückwirkend verfolgen zu können, ist vielmehr auch eine Erfassung der Entwicklung einzelner Modellbestandteile notwendig. Die Protokollierung der Modifikationen wird entsprechend auf Ebene der Objekte angestrebt (vgl. [Somm95], S. 676). Veränderungen an Modellbestandteilen erfordern eine Evaluierung und Planung im Vorfeld sowie eine anschließende Dokumentation, um unerwünschte Seiteneffekte zu verhindern und eine zukünftige Weiterentwicklungen zu ermöglichen.

Daneben dürfen eingebrachte Änderungen nicht die Konsistenz der verwalteten Objekte beeinträchtigen, womit zusätzliche Maßnahmen zur Konsistenzsicherung bzw. -wahrung notwendig werden. Um diese Ziele realisieren zu können, ist ein umfassenderer Ansatz erforderlich. Er wird durch das Konfigurationsmanagement (KM) bereitgestellt. Dessen Ziel ist es „... die gegenwärtige Konfiguration eines Produkts sowie den Stand der Erfüllung seiner physischen und funktionellen Forderungen zu dokumentieren und volle Transparenz herzustellen. Ein weiteres Ziel ist, dass jeder am Projekt Mitwirkende zu jeder Zeit des Produktionslebenslaufs die richtige und zutreffende Dokumentation verwendet.“ ([DIN96], S. 7). Ähnliche Definitionen, allerdings zumeist auf die Softwareentwicklung bezogen, finden sich beispielsweise bei CONRADI und WESTFECHTEL (vgl. [CoWe96], S. 3) und GIESEL (vgl. [Gies98], S. 32). In diesem Beitrag wird unter einem Konfigurationsmanagement-System (KMS) für Modelle (Modell-KMS) sowohl die zu entwickelnde Konzeption, als auch deren Realisierung in einem konkreten System verstanden.

Den zentralen Bestandteil eines KM bildet die Konfiguration zur Abbildung funktionaler und physischer Produktmerkmale in technischen Dokumenten. Sie setzt sich aus Versionen der im KMS verwalteten Objekte zusammen, welche als Konfigurationseinheiten bezeichnet werden (vgl. [DIN96], S. 5f). Ihre verschiedenen Zustände bzw. auch Versionen der Konfiguration stellen Konfigurationselemente dar, wie es insb. in den neueren Ansätzen von CONRADI und WESTFECHTEL vertreten wird (vgl. auch [CoWe97] und [CoWe96], S. 14f). Damit kann eine einheitliche Behandlung aller in einem KMS verwalteten Elemente erfolgen. Versionen eines Objekts stehen über eine, später im Detail zu bestimmende, Menge an gemeinsamen Eigenschaften, der Invariante, zueinander in Beziehung (vgl. auch [CoWe96], S. 8; [West91], S. 57 und [Rose92], S. 8). Die Gesamtheit aller Versionen eines Objekts bzw. einer Konfigurationseinheit wird als Versionsfamilie bezeichnet.

2 Anforderungsanalyse

Gegenstand der folgenden Abschnitte ist die Erarbeitung zu berücksichtigender Anforderungen an das zu konzipierende Modell-KMS. Hierzu wird in Abschnitt 2.1 zunächst die Grundlage für die weiteren Betrachtungen geschaffen. Anschließend untersucht Abschnitt 2.2 allgemeingültige Anforderungen an ein KMS. Die Ergebnisse werden in Abschnitt 2.3 für das hier relevante Gebiet der Modelle präzisiert und erweitert.

2.1 Grundlage der Anforderungsanalyse

Der Erfolg eines KMS hängt nicht nur maßgeblich von seinem Einsatz im Projekt, sondern insb. von der Umsetzung der Verfahren durch die Projektmitarbeiter ab. Voraussetzung dafür ist ein hoher Akzeptanzgrad durch eine echte Unterstützung des Entwicklungsprozesses bei minimaler Behinderung der Arbeiten (vgl. auch Silver Bullet Anti-Pattern ([Brow+99], S. 63ff) in Verbindung mit P2 AntiPattern ([Brow+99], S. 179)). Somit ist die Erfassung der Anforderungen potentieller Benutzer in den Mittelpunkt der folgenden Untersuchungen zu stellen. Um in der Analyse von einer konkreten Organisationsstruktur abstrahieren zu können, wird das Konzept der organisatorischen Rolle verwendet, wie es KIESER und KUBICEK definieren (vgl. [KiKu83], S. 397). Mit den einzelnen Rollen verknüpfte Ziele und Aufgaben bilden folglich die Basis der Anforderungen an ein KMS, und stellen somit den Ausgangspunkt zur Bestimmung der notwendigen Strukturen und Operationen dar. Als Grundlage des hier verwendeten Rollenmodells werden Erfahrungen des Software Engineering Institute (Carnegie Mellon University) in der Softwareentwicklung herangezogen (vgl. [Dart91], S. 2f und [Brow+91], S. 4f), die für die Zwecke dieser Arbeit generalisiert wurden. Demzufolge sind die Aufgaben und Ziele der Rollen Projektmanager, KM-Verantwortlicher, Entwickler, Tester, Qualitätsbeauftragter und Kunde im KMS zu unterstützen (vgl. [Dart91], S. 2f).

Die Strukturierung potentieller Benutzer dient gleichzeitig auch als Basis zur Vermeidung von Interessenskonflikten zwischen den Projektbeteiligten. Aufgrund ihrer unterschiedlichen Aufgaben und Ziele benötigen sie jeweils nur eine Teilmenge der verfügbaren Funktionalität. Die erforderlichen Zugriffsbeschränkungen können mit den Rollen verknüpft werden, und bilden somit den Ausgangspunkt zur Schaffung eines Berechtigungskonzepts innerhalb der Entwicklungsumgebung.

2.2 Allgemeine Anforderungen an ein KMS

Zur Untersuchung allgemeiner Anforderungen an ein KM sind im Folgenden grundlegende und ergänzende Erfordernisse zu unterscheiden. Erstere werden durch die Bestimmungen von DIN 10007 (vgl. [DIN96]) gebildet, die das Minimum abzudeckender Funktionen in einem KM darstellen. Ergänzungen resultieren primär aus den KM-Erfahrungen in der Softwareentwicklung. Ihre Umsetzung vermeidet übermäßige Behinderungen des Entwicklungsprozesses durch den erhöhten Verwaltungs- und Dokumentationsaufwand.

Grundlegende Anforderungen nach DIN EN ISO 10007

Die Bestimmungen der Norm DIN EN ISO 10007 zum Konfigurationsmanagement identifizieren vier grundlegende Bereiche eines KM (vgl. [DIN96], S. 7f):

- Konfigurationsidentifikation (KI): Bestimmung zu verwaltender Elemente (Konfigurationseinheiten, Dokumente) und Erstellung verbindlicher Produktversionen
- Konfigurationsbuchführung (KB): Sicherstellung der erforderlichen Dokumentation sowohl bez. des notwendigen Umfangs als auch der Aktualität (besitzt Dienstleistungscharakter für andere Funktionsbereiche, Grundlage des Konfigurationsaudits)
- Konfigurationsüberwachung (KÜ): Maßnahmen zur Überwachung der Änderungen an einer Konfigurationseinheit (Change Request (CR) Verfahren); DIN EN ISO 10007 ergänzt Bewertung, Koordination, Genehmigung oder Ablehnung von Änderungen
- Konfigurationsaudit (KA): funktionsbezogener KA als Abgleich des Produkts mit der zugrunde liegenden Anforderungsspezifikation (für Modelle aufgrund der subjektiven Wahrnehmung nicht direkt anwendbar); physischer KA als Gegenüberstellung der aktuellen Konfiguration mit den vorhandenen Konfigurationsdokumenten

Neben der Abdeckung dieser Teilgebiete zeichnet sich ein umfassendes KMS insb. auch durch deren Integration aus. Sie wird durch einen zusätzlichen Bereich KM-Planung und Organisation übernommen, der in Abhängigkeit vom bearbeiteten Projekt zu gestalten ist (vgl. [Sayn98], S. 13). Die hier zu erarbeitende Konzeption kann jedoch lediglich Ansatzpunkte für die Integration der Teildisziplinen spezifizieren, deren Nutzung durch den Bereich der Organisation und Planung projektabhängig zu entscheiden ist.

Der Begriff des Audit ist nicht allein auf das verwaltete Produkt zu beschränken, sondern auch auf das KMS zu erweitern. Neben der Wirksamkeit implementierter Abläufe und Strukturen ist insb. auch die Übereinstimmung der angewandten Verfahren mit der Dokumentation des KMS zu überprüfen (vgl. [DIN96], S. 18).

Ergänzende Anforderungen aus Best Practices

Erfahrungen aus der Softwareentwicklung zeigen, dass die aus der DIN Norm ableitbaren KM-Aspekte zumindest um die Bereiche Prozessunterstützung, Teamwork und Buildunterstützung zu erweitern sind (vgl. auch [Dart91], S. 3ff; [Dart92], S. 2 und [Brow+91], S. 5ff):

- Unterstützung des Entwicklungsprozesses: Erreichung einer hohen Akzeptanz der durch das KMS bedingten, unvermeidlichen Zusatzbelastungen bei den betroffenen Entwicklern (Voraussetzung zur Einhaltung von Bestimmungen und Verfahren des KM)
- Unterstützung des Teamworks: KMS als Kommunikationsbasis zur Überprüfung von Modellanforderungen und der Mittel zu ihrer Überprüfung (z. B. zur Vermeidung von Fehlern dritter Art (vgl. [Gait79], S. 8))
- Unterstützung des Build: Erstellung weiterer Abbildungen aus den Modellen (z. B. Erzeugen von Quellcode aus UML-Klassenstrukturdiagramm)

2.3 Anforderungen an ein KMS aus Modellsicht

Nachdem zunächst allgemeine Anforderungen an ein KMS erarbeitet wurden, stehen nun notwendige Ergänzungen und Erweiterungen zur Verwaltung von Modellen im Mittelpunkt der Betrachtungen. Hierbei sind der Modellinhalt und seine Veränderungen zu erfassen sowie prozessuale und organisatorische Anforderungen zu berücksichtigen. Für ersteres sind drei Komponenten zu betrachten: Modellelemente zur Darstellung der wahrgenommenen Entitäten des Originals, der Problembezug und die Namenskonvention.

Gemäß den Ausführungen von WAND (vgl. [Wand89], S. 540f) werden Entitäten des Originals aufgrund bestimmter Eigenschaften wahrgenommen. Hierbei ist es jedoch nicht erforderlich, alle zu erfassen (Konzept der Präterition bei STACHOWIAK, vgl. [Stac73], S. 155). Die Darstellung der wahrgenommenen Eigenschaften erfolgt durch Zuordnung entsprechender Attribute zu den Modellelementen.

Zieht man des Weiteren die Darlegungen STACHOWIAKs hinzu (vgl. [Stac73], S. 155), so bilden Modellelemente „... Merkmale und Eigenschaften von Individuen, Relationen zwischen Individuen, Eigenschaften von Eigenschaften, Eigenschaften von Relationen usw. ...“ ([Stac73], S. 134) ab. Die Entscheidung, ob beliebige Elemente Attributfunktion erfüllen oder als Individuen fungieren, ist vom jeweils bearbeiteten Zusammenhang abhängig. Für die Verwaltung von Modellen im KMS bleibt somit festzuhalten, dass geeignete Strukturen zur Ablage von Modellelementen (inkl. ihrer Attribute) und deren Beziehungen zu berücksichtigen sind.

Ein zentraler Bestandteil des hier verwendeten Modellbegriffs wird durch den Problembezug gebildet, der die Realisierung des pragmatischen Modellmerkmals sicherstellt

(vgl. auch Abschnitt 1.1). Er dient der Erfassung von Anforderungen an das zu erstellende Modell, und stellt daher einen wichtigen Vertragsbestandteil zwischen dem Auftraggeber und den Modellentwicklern dar. Probleme entstehen jedoch während seiner Erarbeitung, da der Anwender zu Beginn des Projekts nicht „... genau weiß, was er will ...“ bzw. keine Vorstellungen über die potentiellen Möglichkeiten hat (vgl. [DaRa97], S. 5).

Die Zusammenstellung des Problembezugs wird des Weiteren zumeist durch das Fehlen einer Kommunikationsbasis, also eines gemeinsamen Begriffsverständnisses erschwert. Aus diesem Grund ist zusätzlich eine Namenskonvention im Projekt zu führen, die den verwendeten Basiswörtern die jeweils zutreffende Definition bzw. Semantik zuordnet. Allerdings greift die alleinige Verwaltung von Begriffsdefinitionen zu kurz.

Im Rahmen der allgemeinen Anforderungen an KMS wurde bereits die Notwendigkeit zur Unterstützung von Teamarbeit hervorgehoben. Übertragen auf den hier vorliegenden Fall der Modellverwaltung bedeutet dies, dass einzelne Entwickler an Teilmodellen arbeiten, die abschließend miteinander integriert werden. Hierbei sollte aber nicht nur eine Auswahl, sondern eine echte Verschmelzung der Arbeitsergebnisse möglich sein. In diesem Rahmen treten gemäß den Darlegungen von HARS (vgl. [Hars93], S. 176ff) und ROSEMANN (vgl. [Rose96], S. 187ff) Namens-, Typ- und Strukturkonflikte auf. Voraussetzung zur Lösung ersterer ist die Vermeidung von Homonymie und Synonymie Sprachdefekten (vgl. [Rose96], S. 187). Während sich letztere durch Verwaltung entsprechender Synonymlisten zu den einzelnen Basiswörtern vermeiden lassen, kann das Homonymieproblem nur durch geeignete Regeln zur Erstellung und Anwendung der Namenskonvention behoben werden.

Wie bereits Abschnitt 2.2 darlegte, sind gewünschte Veränderungen am Modell durch entsprechende Change Requests in den Entwicklungsprozess einzubringen. Sie bedürfen somit einer geeigneten Dokumentation und Verwaltung im KMS. Um zusätzlich auch die Entwicklung einzelner Komponenten nachvollziehen zu können, sind außerdem Dokumentationen einzelner Versionen, sowohl des Modells als auch seiner Bestandteile, in Form von Versionsdokumenten zu führen.

Aus prozessualer Sicht erfordert die Modellentwicklung, neben der Bereitstellung geeigneter Verwaltungs- und Modellieroperationen, insb. auch eine Unterstützung der Zusammenarbeit zwischen Anwender und Entwickler, bzw. zwischen den Entwicklern untereinander. Dieses Gebiet wird unter den Begriff der kollektiven Modellbildung zusammengefasst (vgl. [Herr91], S. 326ff und [Schü98], S. 61).

Erstere erfordert eine Ausrichtung der individuellen Konstruktionsleistungen der Modellierer auf die Ziele des Anwenders. Das hierzu notwendige mutual understanding zwischen Anwender und Entwickler (vgl. [Herr91], S. 328f) entsteht jedoch nur bei Kenntnis der jeweiligen psychologischen Typologien. Dieser Prozess kann deshalb nicht durch das Modell-KMS unterstützt werden.

Die Kooperation der Modellersteller untereinander erfordert primär die möglichst optimale Unterstützung der verschiedenen Vorgehensweisen aufgrund der individuellen Ausprägungen von Kenntnissen, Psyche und Wertesystem. Dies bedingt private Arbeitsbereiche, im Folgenden als Workspaces bezeichnet, die den jeweiligen Entwickler vor Manipulationen anderer schützen, gleichzeitig aber auch seine eigenen Änderungen auf den geschützten Bereich beschränken.

Für die Erstellung des Gesamtmodells aus den einzelnen Arbeitsergebnissen sind zusätzlich Verfahren zur kontrollierten Integration von Teilmodellen vorzusehen. Organisatorische Anforderungen durch die Anwendung des in Abschnitt 2.1 entwickelten allgemeinen Rollenmodells wird insb. dem Kunden eine weitere Spezialisierung zuteil. Der durch sie repräsentierte Modellanwender kann sowohl unternehmensinternen als auch -externen Charakter besitzen. Hieraus leiten sich wiederum Konsequenzen für die erforderlichen Test- und Qualitätssicherungsmaßnahmen ab. So wird u. U. in unternehmensinternen Modellierungsprojekten der Entwicklungsgeschwindigkeit eine höhere Priorität als der Qualitätssicherung eingeräumt. Dementsprechend sind im KMS Optionen vorzusehen, die eine Anpassung der Konzeption an die Gegebenheiten des aktuellen Projekts bzw. des einsetzenden Unternehmens ermöglichen.

3 Spezifikation eines Modell-KMS

Ziel der Ausführungen in den folgenden Abschnitten ist die Entwicklung einer Konzeption zum KM in Modellen. Hierzu wird weiterhin die in der Anforderungsanalyse verwendete Unterteilung nach Struktur-, Organisations- und Prozesssicht eingesetzt. Abschließend erfolgt die Integration der einzelnen Teilsichten, um eine ganzheitliche Unterstützung der Modellbildung entsprechend der Anforderungen aus Kapitel 2 zu ermöglichen. Das Kapitel schließt mit der Darstellung des Prozesses zum Einbringen von Change Requests in ein Modellierungsprojekt.

Aufgrund des begrenzten Umfangs dieses Beitrags werden im Folgenden nur ausgewählte Schwerpunkte der Spezifikation dargestellt. Für eine ausführliche und umfassende Darstellung wird auf die Dokumentation des auf Basis dieses Ansatzes entwickelten Modellierungswerkzeugs verwiesen (vgl. [Grei02]).

3.1 Strukturelle Spezifikation

Konfigurationseinheitstypen

Aufgrund des in Abschnitt 1.2 eingeführten identischen Behandlung aller Konfigurationseinheiten werden deren Gemeinsamkeiten in einem Supertypen **Version** zusammengefasst. Zur Abbildung der individuell unterschiedlichen fachlichen Anforderungen an die Konfigurationseinheitstypen werden daraus entsprechende Subtypen spezialisiert. Aus den bereits oben erwähnten Platzgründen wird auf die detaillierte Betrachtung der Typen für den Problembezug und die Namenskonvention verzichtet.

Version

Bevor die strukturelle Spezifikation der Version durch Zuordnung der notwendigen Attribute vervollständigt werden kann, sind zunächst grundsätzliche Fragen zur Speicherung und Identifikation von Versionen zu klären. Für diesen Beitrag wird das zustandsorientierte Versionisierungsmodell eingesetzt (vgl. [CoWe96], S. 8), das stets die vollständige physische Speicherung einzelner Versionen vorsieht. Es bietet gegenüber dem ebenfalls möglichen änderungsorientierten Konzept den Vorteil einer vereinfachten Umsetzung, bei gleichzeitiger Abdeckung aller relevanten Anforderungen.

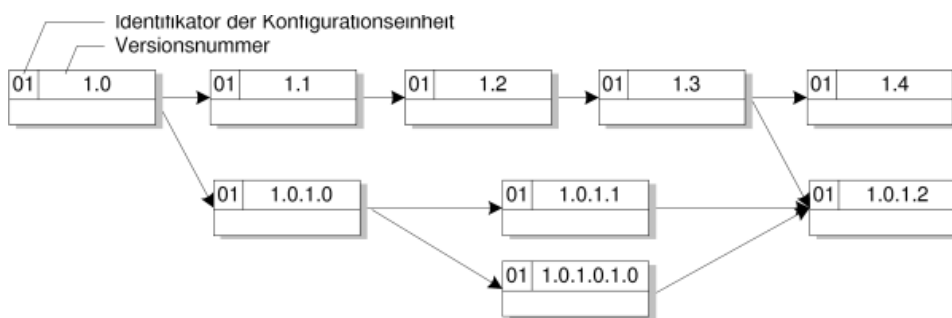


Abbildung 1: Beispiel zum Versionsnummernkonzept

Zur Identifikation von Versionen einer Konfigurationseinheit wurde in Abschnitt 1.2 die Invariante eingeführt. Sie lässt sich jedoch für die unterschiedlichen Arten von Modellelementen erst nach der Auswahl des jeweiligen Meta-Modells festlegen. Um eine möglichst universell einsetzbare, aber dennoch umfassende Spezifikation zu ermöglichen, werden hier KEs einer Versionsfamilie über einen gemeinsamen und systemweit eindeutigen Identifikator unterschieden. Zur Differenzierung und Einordnung von KEs innerhalb einer Versionsfamilie wird eine Versionsnummer verwandt (extensionales Konzept, vgl. [Ask1+99], S. 102ff). Hierfür kommt das Concurrent Versioning System (CVS) zum Einsatz (vgl. [Cede98], S. 7f). Ein Beispiel hierfür zeigt die Abbildung 1.

Modellelement

Gemäß den Darlegungen in Abschnitt 2.3 werden die Entitäten des Originals über deren Eigenschaften wahrgenommen. Sie werden in Form von Attributausprägungen des korrespondierenden Modellelements in das Modell eingebracht. Die Festlegung der möglichen Attribute eines Modellelements erfolgt durch seinen Typen im zugehörigen Meta-Modell.

Wird des Weiteren das Meta Object Facility (MOF) (vgl. [Obj00]) zugrunde gelegt, so kann das Meta-Modell ebenfalls als Modell, allerdings auf MOF-Ebene M2, aufgefasst werden. Es stellt im Modell-KMS somit auch eine Konfiguration dar, die in die aktuelle Modellversion integriert werden kann. Die Ablage der Beziehung zwischen einem Modellelement und seinem Typen erfolgt in der Konfigurationsstruktur. Mithin ist für die

Gültigkeit eines Modells im Modell-KMS zu fordern, dass jedes Modellelement genau eine Beziehung zu einem Modellelement des zugehörigen Meta-Modells besitzt. Um Modelle der obersten geplanten Meta-Ebene erfassen zu können, müssen stets verfügbare Basistypen eingeführt werden, die nicht der Versionierung unterliegen. Ihre Definition erfordert keine Beziehung zu einem übergeordneten Typen. Aufgrund dieser Besonderheiten werden sie einer speziellen Konfiguration verwaltet.

Konfigurationsmanagementplan

Der Konfigurationsmanagement-Plan (KMP) legt für ein spezifisches Produkt oder Projekt die KM-Organisation und die anzuwendenden Verfahren fest (vgl. [DIN96], S. 6). Er enthält die Beschreibung der anzuwendenden Verfahren, zu ihrer Durchführung berechtigter Projektmitarbeiter und ihrer Rollen sowie des Durchführungszeitpunkts (vgl. [DIN96], S. 17). Gleichzeitig wird aber auch seine anpassende Funktion deutlich, da mit den Festlegungen sowohl eine Auswahl aus den verfügbaren Verfahren erfolgt, als auch das KMS insgesamt auf die Projektanforderungen abgestimmt wird. Somit bilden die Informationen des KMP die Grundlage zum Audit des Modell-KMS. Diesbezügliche Erfordernisse werden allerdings maßgeblich vom Kunden beeinflusst, so dass der KMP neben dem Problembezug ebenfalls einen wichtigen Vertragsbestandteil darstellt. Obwohl seine Aufgabe somit primär dokumentierenden Charakter trägt, wird er trotzdem als Konfigurationseinheit behandelt. Damit können zu jedem Modellstand die aktuell zutreffenden KM-Bestimmungen verwaltet werden.

Seine Attributierung lehnt sich an die Empfehlungen der DIN Norm (vgl. [DIN96], S. 19f) an, die neben den KM-Funktionsbereichen (KI, KÜ, KB und KA) auch Abschnitte zu allgemeinen Grundsätzen und Verfahren, sowie zu allgemeinen Projektinformationen vorsieht. Wie Beispiele aus der Praxis zeigen (vgl. [US D00] und [Info97]), hat sich eine derartige Strukturierung bewährt. Da auch die DIN Norm lediglich Empfehlungen zu den einzelnen Abschnitten des KMP enthält, wird hier ebenfalls nur eine Attributierung auf Kapitelebene vorgeschlagen, deren Ausgestaltung im Detail den Gegebenheiten im aktuellen Projekt überlassen bleibt.

Konfiguration

Gemäß den Ausführungen des Abschnitts 1.2 werden die zu einem Modell zugehörigen Versionen der Modellelemente, des Problembezugs, der Namenskonvention, des KMP und der Konfigurationsstruktur zentral durch Versionen der Konfiguration verwaltet. Sie können damit begrifflich Modellversionen gleichgesetzt werden. Zu ihrer Integration in die KM-Konzeption wird ein entsprechender Konfigurationseinheitstyp **Configuration** eingeführt.

Da für jedes Modellelement im hier entwickelten Konzept ein Typelement notwendig ist, muss eine Konfigurationsversion zwingend eine Version einer weiteren Konfiguration enthalten, die das verwendete Meta-Modell repräsentiert. Kommen als Typen die definierten Basistypen zum Einsatz, so wird als Meta-Modell die spezielle Konfiguration C_0 verwendet. Sie liegt stets in der initialen Version vor und enthält ausschließlich Basistypen.

Aufgrund der Heterogenität der verwalteten Konfigurationseinheiten kann die Invariante der Konfiguration nicht eindeutig bestimmt werden. Die Frage, ob die Ausführung einer Operation zur Erstellung eines neuen Modells oder lediglich zur Bildung einer neuen Version führt, lässt sich nur bei gleichzeitiger Betrachtung der Operation und des Typs der betroffenen Konfigurationseinheit beantworten. So sollte das Einfügen eines Modellelements zur Bildung einer Modellversion führen. Wird dagegen der Problembezug aktualisiert, muss ein neues Modell erzeugt werden, da sich die Ziele und Anforderungen der Anwender verändert haben. Damit wird im Prinzip ein neues Projekt begründet. Die zusätzlichen Restriktionen sind im Rahmen der Integration der Teilsichten (vgl. Abschnitt 3.4) entsprechend zu berücksichtigen.

Konfigurationsstruktur

Wie in Abschnitt 2.3 bereits erarbeitet wurde, erfordert die Verwaltung von Modellen nicht nur die Erfassung der Modellelemente, sondern auch der Beziehungen zwischen diesen. Zur Verwaltung letzterer wird der Konfigurationseinheitstyp **ConfigurationStructure** in das Konzept aufgenommen. Da die Konfiguration alle zugehörigen KEs direkt verwaltet, ist eine separate Behandlung der Beziehungen zwischen ihnen und der jeweiligen Modellversion nicht erforderlich. Mithin reduzieren sich die in der Konfigurationsstruktur erfassten Verknüpfungen auf:

- Typbeziehungen – Beziehungen zwischen Basistypen/Modellelementversionen aus der Meta-Modellversion und Modellelementversionen im bearbeiteten Modell
- Modellinterne Beziehungen – Beziehungen zwischen Modellelementversionen innerhalb des bearbeiteten Modells

Aufgrund der unterschiedlichen Verknüpfungsemantiken gestaltet sich die Festlegung der Invariante ähnlich komplex wie im Falle der Konfiguration. Werden modellinterne Beziehungen entfernt, so handelt es sich lediglich um eine Weiterentwicklung des Modells, die durch die Erstellung einer neuen Version der Konfigurationsstruktur repräsentiert wird. Wird dagegen eine Typbeziehung verändert, so ändert sich gleichzeitig auch der Charakter des betroffenen Modellelements. Dies ist zwingend durch die Erstellung einer neuen Konfigurationseinheit für das Modellelement zu repräsentieren.

Konfigurationsdokumenttypen

Aufgrund der Anforderungen aus der DIN Norm 10007 lassen sich zwei Dokumenttypen identifizieren. Zum einen dient das Versionsdokument zur Beschreibung einzelner KEs, während Change Request Dokumente (CRD) die notwendigen Informationen zu einem eingebrachten Änderungsantrag verwalten. Allen Dokumenten ist jedoch gemeinsam, dass sie nicht versioniert werden, da sie der Beschreibung eines Elements im KMS dienen.

Aufgabe des Versionsdokuments ist es, insb. die Erstellungsgründe und Eigenschaften eines KE zu beschreiben. Ferner werden auch die für seine Entwicklung verantwortli-

chen Mitarbeiter erfasst, um später eine genaue Analyse der Entwicklung zu ermöglichen.

Im Gegensatz dazu dokumentiert das CRD, neben den Gründen und Zielen einer beantragten Änderung, auch ihren Genehmigungsstatus sowie von der Realisierung betroffene Konfigurationseinheiten. Außerdem werden ebenfalls die Ergebnisse der Änderungsrealisierung, der verursachte Aufwand und die jeweils beteiligten Mitarbeiter beschrieben.

3.2 Organisationale Spezifikation

Mit der Analyse der zu unterstützenden Benutzergruppen in Abschnitt 2.1 wurden Rollen identifiziert, die sowohl lang- als auch kurzfristige Interessen des Unternehmens verfolgen. Aufgabe der Projektleitung ist es, das aktuelle Modellierungsprojekt sowohl im Budget- als auch Zeitrahmen fertig zu stellen. Sie verfolgt damit primär kurzfristige Ziele. Im Gegensatz dazu dienen die Tätigkeiten der Tester und Qualitätssicherung im Wesentlichen der Wahrung langfristiger Ziele, wie z. B. der Möglichkeit der Weiterentwicklung des Produkts.

Um die konträren Interessen im Projekt aufeinander abzustimmen, können keine Hierarchiebeziehungen zwischen den Gruppen eingeführt werden. Dies hätte stets eine potentielle Benachteiligung der jeweils untergeordneten Rollen zur Folge. Stattdessen wird im Modell-KMS ein integrierendes Gremium, das Change Control Board (CCB) eingeführt. Es umfasst Mitglieder aller relevanten Benutzergruppen. Seine Aufgabe besteht neben dem Ausgleich der konfligierenden Interessen insb. auch in der Kontrolle von einzubringenden Änderungen am Produkt. Es dient somit gleichzeitig auch als zentrale Genehmigungsinstanz für Change Requests.

3.3 Funktionale Spezifikation

Mit den folgenden Ausführungen werden die zur Verwaltung und Entwicklung von Modellen im KMS erforderlichen Operationen dargestellt. Sie lassen sich grob in die folgenden Bereiche unterteilen:

- KM: Verwaltung verschiedener Zustände von Modellen und ihrer Elemente
- Modellierung: Hinzufügen bzw. Entfernen von KEs und Beziehungen zwischen diesen
- Berichte: Dokumentation der Änderungen

Aufgrund ihrer Bedeutung für die Abwicklung der Modellerstellung konzentrieren sich die folgenden Darstellungen auf KM- und Modellieroperationen. Die Berichtsoperationen besitzen eine starke Abhängigkeit von den Anforderungen des Projekts, die DIN 10007 gibt lediglich Anhaltspunkte zu ihrer Ausgestaltung (vgl. [DIN96], S. 16f). Entsprechend sind die Berichte auf Basis aktueller Benutzeranforderungen zu erstellen.

Die zu einem Modell-KMS gehörenden Workspaces stehen untereinander in einer hierarchischen Beziehung, d. h. jeder Arbeitsbereich besitzt genau einen übergeordneten. Die Spitze der Hierarchie wird durch einen speziellen Workspace, das Repository gebildet. Er ist stets vorhanden, benötigt keinen übergeordneten Arbeitsbereich und dient primär der Archivierung erstellter und ausgelieferter Modellversionen. Die folgenden Operationen werden in den Workspaces definiert:

Operation	Beschreibung	Beispiel der Anwendung	Unterarten
CheckOut	Übertragung einer Modellversion in einen untergeordneten Workspace in Form einer Konfiguration	Parallele Bearbeitung und Verteilung von Modellversionen	Export
CheckIn	Übertragung einer Modellversion in einen übergeordneten Workspace in Form einer Konfiguration	Zusammenführung der parallel erstellten Modellversionen	Import Merge Branch
Import	Integration von Modellversionen, die keine vorhergehende Version im übergeordneten Workspace besitzen	Vom Repository losgelöstes Arbeiten an einem Modell	
Export	Transformation einer Konfigurationsversion in einen Ziel-Workspace mit Schutz vor Änderungen	Für Meta-Modell-Versionen oder an Kunden ausgelieferte Modelle	
Merge	Verschmelzung zweier Modellversionen aus verschiedenen Zweigen des Versionsgraphen, die auf einer gemeinsamen Ausgangsversion beruhen	Einbringung eines Modellstands als Nachfolgeversion im übergeordneten Workspace, wenn Arbeiten am Modell seit dem CheckOut erfolgten	
Branch	Eröffnen eines neuen Entwicklungspfades eines Modells	Keine Verschmelzung der Ergebnisse sondern gleichberechtigte Modellvarianten	
Lend	Austausch von KEs (außer Configuration und ConfigurationStructure)	Übertragung von Modellelementen in beliebige Modelle	

Tabelle 1: Operationen im Workspace

Für die einzelnen KE werden die folgenden Modellieroperationen definiert:

Operation	Beschreibung	Beispiel der Anwendung	Eigentümer
CreateModel	Erzeugen einer Konfiguration und Konfigurationsstruktur	Erstellung einer initialen Konfiguration	Workspace
AddElement	Hinzufügen eines KE	Modellierung zusätzlicher Inhalte	Configuration
Remove-Element	Entfernen eines KE	Löschen von Informationen	Configuration
Connect-Elements	Definition einer Beziehung zwischen zwei Elementen	Verbinden von zwei Elementen	Configuration-Structure
Disconnect-Elements	Löschen einer Beziehung zwischen zwei Elementen	Trennen von zwei Elementen	Configuration-Structure

Tabelle 2: Operationen auf Konfigurationselemente

Aufgrund der identischen Behandlung von Modell und Meta-Modell als Konfigurationen im Modell-KMS ist es auch möglich, einen Austausch des Meta-Modells in der Entwicklung vorzunehmen. Da hiermit der Charakter des Modells nachhaltig verändert wird, kann dieser Schritt nur durch die Erstellung eines neuen Modells geeignet repräsentiert werden. Es entspricht dem Stand der zugrunde liegenden Modellversion, verwendet aber bereits das neue Meta-Modell. Modellelemente, deren Typen im neuen Meta-Modell nicht mehr vorhanden sind, und denen kein neuer Typ durch den Modellierer zugewiesen werden kann, werden nicht in das neue Modell übernommen.

3.4 Integration der Teilspezifikationen

Mit den Ausführungen zur strukturellen, organisatorischen und funktionalen Spezifikation wurden die notwendigen Grundlagen zur Verwaltung von Modellen in verschiedenen Entwicklungszuständen geschaffen. Die Anforderungsanalyse des Abschnitts 2 forderte aber auch eine Unterstützung des Entwicklungsprozesses durch das KMS. Ein fest definiertes Vorgehensmodell steht jedoch im Gegensatz zur notwendigen Kreativität im Modellbildungsprozess (vgl. Abschnitt 2.3). Um diese konträren Forderungen möglichst optimal unterstützen zu können, wird auf die Einbeziehung eines Vorgehensmodells verzichtet. Die prinzipielle Verfügbarkeit aller Strukturen und Funktionen KMS-Benutzern wird lediglich durch folgende drei Komponenten eingeschränkt:

- Typ der bearbeiteten KE: Sicherstellung der korrekten und sinnvollen Anwendung der KMS-Funktionen
- Entwicklungsstand der KE: Einhaltung zwingender Meilensteine im Projekt (z. B. nach Testen und Warten müssen evtl. Veränderungen ausgeschlossen werden)
- Zugriffsrechte des aktuellen Benutzers: Verknüpfen von Ausführungsrechten mit den Rollen im KMS zur Sicherung der Kompetenzen und Verantwortlichkeiten

Entsprechend werden vom KMS für einen bestimmten Zeitpunkt die Operationen wie aus Abbildung 2 ersichtlich eingeschränkt.

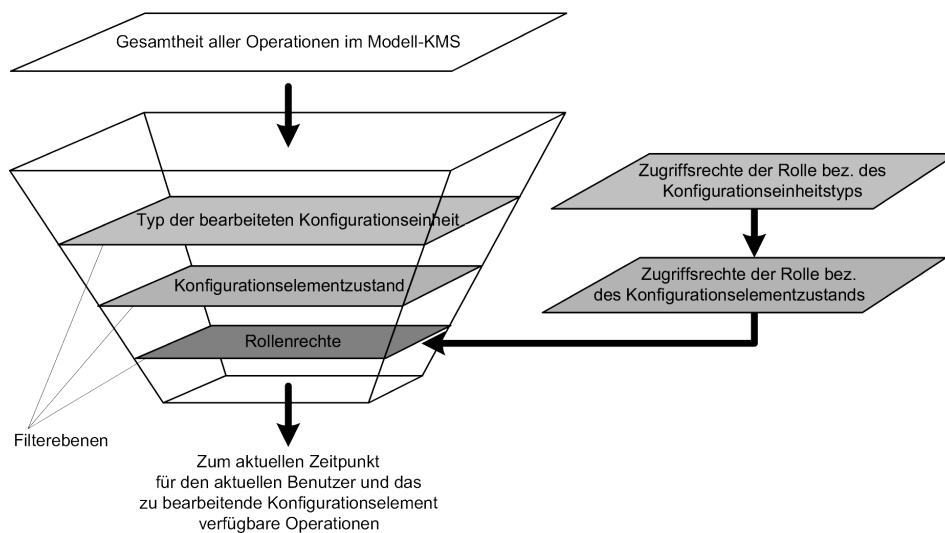


Abbildung 2: Verfügbare Operationen im KMS

Die Integration des Ansatzes in das KMS erfolgt durch die Einführung eines Zustandsmodells für KEs, deren Zustände die einzelnen Entwicklungsschritte (Erstellung, Entwicklerfreigabe, Test, Qualitätssicherung, Auslieferung) widerspiegeln. Die detaillierte Beschreibung der Verknüpfung von Zuständen und zulässigen Operationen kann in diesem Rahmen jedoch nicht aufgeführt werden.

3.5 Change Request Prozess

Mit Hilfe der bisher dargestellten Strukturen, Operationen und Integrationsansatzes ist eine Verwaltung von Modellen und deren Veränderungen inkl. der erforderlichen Dokumentation sichergestellt. Allerdings ist entsprechend den Ausführungen aus Abschnitt 2 nicht nur die Aufzeichnung, sondern auch die Kontrolle von Änderungen von Bedeutung. Somit sind Change Requests vor ihrer Realisierung zunächst zu analysieren und anschließend durch eine zentrale Instanz zu genehmigen. Dieses Vorgehen in allen Projektphasen würde jedoch nicht den alternierenden Erfordernissen im Zeitablauf gerecht werden. So sind z. B. Veränderungen in den frühen Phasen der eigentlichen Modellentwicklung bzw. Abstimmungen zwischen den Erstellern geschuldet, und nicht durch wechselnde Anforderungen der Auftraggeber bedingt. Aus diesem Grund wird hier ein adaptierbarer Change Request Prozess (CRP) nach den Ausführungen von PRESSMAN (vgl. [Pres92], S. 706) eingesetzt. Er gliedert das Projekt in drei Phasen, die jeweils unterschiedliche Anforderungen an die Formalisierung von Change Requests stellen:

Phase	Art der Änderungen	Change Request Prozess
Projektstart bis Erstellung einer Bezugskonfiguration (BK)	Primär der Erstellung einer ersten Modellversion geschuldet	Informal (keinerlei Dokumentation und Genehmigung von Änderungen)
Erstellung der BK bis Auslieferung einer ersten Modellversion an den Kunden	Primär durch Entwickler motiviert, dem Zusammenwirken unterschiedlicher Teams auf Basis einer gemeinsamen BK ist Rechnung zu tragen	Änderungsanträge sind mit (Teil-) Projektleiter abzustimmen
Nach Auslieferung einer ersten Modellversion an den Kunden	Primär auf Basis von Änderungs- bzw. Erweiterungswünschen des Kunden	Schriftliche CRDs, Abstimmung im CCB

Tabelle 3: Phasen des Change Request Prozesses

Die Phase des formalen CRP endet erst mit der Einstellung aller Entwicklungsarbeiten am Modell. Die Integration des dargelegten adaptiven CRP in das Modell-KMS erfolgt durch die Ergänzung des im vorigen Abschnitt vorgestellten Zustandsmodells um die hier aufgeführten Phasen (Erstellung, Definition der BK und Auslieferung) für Konfigurationen.

4 Zusammenfassung und Ausblick

Mit dem in diesem Beitrag vorgestellten KMS kann eine Verwaltung und Kontrolle der Entwicklung von Modellen realisiert werden, was am Lehrstuhl für Wirtschaftsinformatik, insb. Systementwicklung der TU Dresden durch die Anwendung der Spezifikation innerhalb einer Methode zur Entwicklung von Anwendungssystemen nachgewiesen werden konnte. Auf deren Basis wird momentan ein Modellierungswerkzeug entwickelt, welches die Konzepte des KM von Modellen praktisch umsetzt (vgl. [Grei02]). Dadurch kann in Zukunft auch eine praktische Evaluation der Akzeptanz dieses Ansatzes bei Anwendern erfolgen und das KMS weiterentwickelt werden.

Nach den Ausführungen von BROWN ET AL. besteht ein umfassender KM-Ansatz aus KM-Primitiven, KM-Protokollen und KM-Strategien (vgl. [Brow+91]). Im Rahmen dieses Beitrages konnten aufgrund der fehlenden praktischen Erfahrungen nur Primitive und Protokolle behandelt werden. Die Strategien bei der Anwendung der Konzepte wird der praktische Einsatz liefern.

Als Hauptkritikpunkte des entworfenen Modell-KMS müssen seine lediglich grundlegende Integration in Umgebungsprozesse (z. B. der einer Systementwicklung), sowie die fehlenden Strategien zu seiner Anpassung an die Erfordernisse unterschiedlicher Projekte genannt werden. Letztere lassen sich durch die Formalisierung des Konzepts bzw. durch den Entwurf von Beispielszenarien für ausgewählte Projektsituationen realisieren. Während sich der durch die Formalisierung entworfene „Customizingbaukasten“ durch ein stets konsistentes und funktionsfähiges System im Ergebnis der Anpassung auszeichnet, bieten die Szenarien den Vorteil einer weniger aufwendigen Erarbeitung und vereinfachten Handhabung.

Literaturverzeichnis

- [Appl00] APPLETON, B.: The ACME Project: SCM Definitions. <http://www.enteract.com/~bradapp/acme/scm-defs.html>, Download: 22.08.2000
- [Ask1+99] ASKLUND, U.; BENDIX, L.; CHRISTENSEN, H. B.; MAGNUSSON, B.: The Unified Extensional Versioning Model. In: ESTUBLIER, J. (Hrsg.): System Configuration Management: 9th International Symposium, SCM-9, Toulouse, France, September 1999, Proceedings. Berlin, Heidelberg: Springer, 1999 (Lecture Notes in Computer Science 1675), S. 100–122
- [Beck+95] BECKER, J.; ROSEMAN, M.; SCHÜTTE, R.: Grundsätze ordnungsmäßiger Modellierung. In: Wirtschaftsinformatik, 37 (1995) 5, S. 435–445
- [Brow+91] BROWN, A.; DART, S.; FEILER, P.; WALLNAU, K.: The State of Automated Configuration Management / Software Engineering Institute (Carnegie Mellon University). Pittsburgh, September 1991 (ATR 92) – Arbeitsbericht. ftp://ftp.sei.cm.edu/pub/case-env/config_mgt/papers/atr_cm_state.pdf, Download: 16.07.2000

- [Brow+99] BROWN, W. J.; "SKIP" MCCORMICK III, H. W.; THOMAS, S. W.: AntiPatterns and Patterns in Software Configuration Management, New York: Wiley, 1999
- [Cede98] CEDERQVIST, P.: Version Management with CVS. <http://wwwinfo.cern.ch/asd/cvs/cvs.ps>, Download: 18.11.2000, 19.08.1998
- [CoWe96] CONRADI, R.; WESTFECHTEL, B.: Version Models for Software Configuration Management / RWTH Aachen, 1996 (AIB 96-10) – Forschungsbericht
- [CoWe97] CONRADI, R.; WESTFECHTEL, B.: Towards a Uniform Version Model for Software Configuration Management. In: CONRADI, R. (Hrsg.): Software Configuration Management: ICSE'97 SCM-7 Workshop, Boston, MA, USA, May 1997. Berlin; Heidelberg: Springer, 1997 (Lecture Notes in Computer Science 1235), S. 1–17
- [DaRa97] DAHME, C.; RAEITHEL, A.: Ein tätigkeits-theoretischer Ansatz zur Entwicklung von brauchbarer Software. In: Informatik-Spektrum, 20 (1997) 1, S. 5–12
- [Dart91] DART, S.: Concepts in Configuration Management Systems. In: SCM 1991: Proceedings of the 3rd International Workshop on Software Configuration Management. Trondheim, Norway: ACM Press, 1991, S. 1–18
- [Dart92] DART, S.: The Past, Present and Future of Configuration Management / Software Engineering Institute (Carnegie Mellon University). Pittsburgh, Juli 1992 (CMU/SEI-92-TR-8) – Forschungsbericht. ftp://ftp.sei.cmu.edu/pub/case-env/config_mgt/tech_rep/cm_past_pres_future_TR08_92.pdf, Download: 13.08.2000
- [DIN96] Deutsches Institut für Normung e.V.: DIN EN ISO 10007 - Qualitätsmanagement: Leitfaden für Konfigurationsmanagement (ISO10007:1995), 1996
- [Dres99] DRESBACH, S.: Epistemologische Überlegungen zu Modellen in der Wirtschaftsinformatik. In: BECKER, J. (Hrsg.); SCHÜTTE, R. (Hrsg.); KÖNIG, W. (Hrsg.); WENDT, O. (Hrsg.); ZELEWSKI, S. (Hrsg.): Wirtschaftsinformatik und Wissenschaftstheorie: Bestandsaufnahme und Perspektiven. Wiesbaden: Gabler, 1999, S. 71–94
- [Estu00] ESTUBLIER, J.: Software Configuration Management: A Roadmap. <http://www.cs.acl.ac.uk/staff/A.Finkelstein/fose/finalestublier.pdf>, Download: 16.07.2000
- [Gait79] GAITANIDES, M.: Konstruktion von Entscheidungsmodellen und 'Fehler dritter Art'. In: Wirtschaftswissenschaftliches Studium, 8 (1979) 1, S. 8–12
- [Gies98] GIESEL, J.: Software-Konfigurationsmanagement. In: [Heil+98], S. 27–41
- [Grei02] GREIFFENBERG, S.: Generischer Modelleditor. <http://wise.wiwi.tu-dresden.de/gme2001>, Download: 8.3.2002
- [Habe93] HABERMANN, H.-J.: Repository: Eine Einführung. München: Oldenbourg, 1993 (Handbuch der Informatik 8.1)
- [Hars93] HARS, A.: Referenzdatenmodelle: Grundlagen effizienter Datenmodellierung. Wiesbaden: Gabler, 1993 (Schriften zur EDV-orientierten Betriebswirtschaft)
- [Heil+98] HEILMANN, H. (Hrsg.); KATZSCH, R. M. (Hrsg.); MEIER, A. (Hrsg.); MEINHARDT, S. (Hrsg.); MÖRIKE, M. (Hrsg.); SAUERBURGER, H. (Hrsg.): Configuration- and Change-Management. Heidelberg: Hüthig, 1998 (HMD – Praxis der Wirtschaftsinformatik 202)
- [Herr91] HERRMANN, H.-J.: Modellgestützte Planung im Unternehmen: Entwicklung eines Rahmenkonzeptes. Wiesbaden: Gabler, 1991 (Neue betriebswirtschaftliche Forschung 89)

- [Info97] INFORMATION SYSTEM DEVELOPMENT SUPPORT (ISDS) TEAM: ISDS Product Assurance: Configuration Management Plan. <http://www-isds.jpl.nasa.gov/cm/html/cm/cmp/isdscmp.htm>, Download: 14.09.2000
- [Jäge82] JÄGER, P. K.: Modellmethodologie und optimale Bestellmenge - Grundriß einer Methodologie der Modellkonstruktion am Modell der optimalen Bestellmenge. Johann Wolfgang Goethe Universität Frankfurt am Main, Dissertation, 1982
- [KiKu83] KIESER, A.; KUBICEK, H.: Organisation. 2. Auflage, Berlin; New York: de Gruyter, 1983
- [Müll80] MÜLLER-MERBACH, H.: Modelldenken und der Entwurf von Unternehmensplanungsmodellen für die Unternehmensführung. In: HAHN, D. (Hrsg.): Führungsprobleme industrieller Unternehmungen: Festschrift für Friedrich Thomée zum 60. Geburtstag. Berlin: de Gruyter, 1980, S. 471–489
- [Moli84] DE MOLIÈRE, F.: Prinzipien des Modellentwurfs - Eine modelltheoretische und gestaltungsorientierte Betrachtung. Technische Hochschule Darmstadt, Dissertation, 1984
- [Obj00] OBJECT MANAGEMENT GROUP: Meta Object Facility (MOF) Specification Version 1.3. <ftp://ftp.omg.org/pub/docs/formal/00-04-03.pdf>, Download: 18.10.2000
- [PoBl93] POMBERGER, G.; BLASCHEK, G.: Software-Engineering: Prototyping und objektorientierte Software-Entwicklung. München; Wien: Hanser, 1993
- [Pres92] PRESSMAN, R. S.: Software Engineering: A Practitioner's Approach. 3. Auflage, New York: McGraw-Hill, Inc., 1992
- [Rose92] ROSE, T.: Entscheidungsorientiertes Konfigurationsmanagement. Berlin; Heidelberg: Springer, 1992 (Informatik-Fachberichte 305)
- [Rose96] ROSEMANN, M.: Komplexitätsmanagement in Prozessmodellen. Wiesbaden: Gabler, 1996 (Schriften zur EDV-orientierten Betriebswirtschaft)
- [Sayn98] SAYNISCH, M.: Grundlagen des Konfigurationsmanagements. In: [Heil+98], S. 7–26
- [Schü98] SCHÜTTE, R.: Grundsätze ordnungsmäßiger Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle. Wiesbaden: Gabler, 1998 (Neue betriebswirtschaftliche Forschung 233)
- [Somm95] SOMMERVILLE, I.: Software Engineering. 5. Auflage, Harlow: Addison-Wesley, 1995
- [Stac73] STACHOWIAK, H.: Allgemeine Modelltheorie. Wien: Springer-Verlag, 1973
- [Stic+97] STICKEL, E. (Hrsg.); GROFFMANN, H.-D. (Hrsg.); RAU, K.-H. (Hrsg.): GABLER Wirtschaftsinformatik-Lexikon. Wiesbaden: Gabler, 1997
- [US D00] US DEPARTMENT OF TRANSPORTATION: Surface Search Radar (SSR) Project: Configuration Management Plan (CMP). <http://www.hjford.com/cg/ssr/Configuration/CMP.html>, Download: 14.09.2000, 2000
- [Wand89] WAND, Y.: A Proposal for a Formal Model of Objects. In: KIM, W. (Hrsg.); LOCHOVSKY, F. H. (Hrsg.): Object-Oriented Concepts, Databases and Applications. New York: ACM Press, 1989, Kapitel 21, S. 537–559
- [West91] WESTFECHTEL, B.: Revisions- und Konsistenzkontrolle in einer integrierten Softwareentwicklungsumgebung. Berlin; Heidelberg: Springer, 1991 (Informatik-Fachberichte 280)
- [Zell97] ZELLER, A.: Configuration Management with Version Sets. Braunschweig, Technische Universität Braunschweig, Dissertation, 22.05.1997. – <http://www.cs.tu-bs.de/softech/papers/zeller-phd/>, Download: 26.06.2000