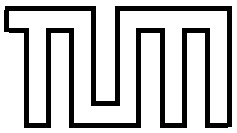
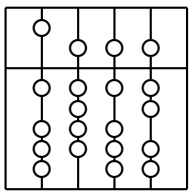


INSTITUT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Diplomarbeit

Konzeptentwicklung für Configuration Management in einem Rechenzentrum nach ITIL und MOF

Bearbeiter:	Florian Sager
Aufgabensteller:	Prof. Dr. Heinz-Gerd Hegering
Betreuer:	Michael Brenner Martin Sailer Volker Leitzgen, Microsoft Deutschland



Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 15. März 2005

.....
(Unterschrift des Kandidaten)

Abstract

Der Prozess des Configuration Managements zum Aufbau und zur Verwaltung einer Configuration Management Database (CMDB), eines logischen Modells über eine IT-Infrastruktur, ist einer der zentralen IT Service Management Prozesse, den die Best Practices Frameworks ITIL (IT-Infrastructure Library) und MOF (Microsoft Operations Framework) beschreiben. Eine CMDB enthält Informationen über Infrastrukturkomponenten mit deren Abhängigkeiten und dient als zentrales Auskunftssystem für andere IT Service Management Prozesse.

Diese Diplomarbeit fasst zunächst die Anforderungen und Problemstellungen bei Einführung und Betrieb von Configuration Management nach ITIL und MOF zusammen und bewertet diese in Bezug auf die Möglichkeiten der Ableitung einer Implementierung. In Folge werden selbst entwickelte und praxisbewährte Lösungsansätze für die effiziente Umsetzung von Configuration Management aufgezeigt, konzeptionell strukturiert und mitunter anhand von Beispielen verdeutlicht. Problemstellung dabei ist unter anderem die Schaffung einer möglichst allgemeingültigen aber umsetzbaren Konkretisierung des Configuration Managements aus den ungenauen Vorgaben in ITIL und MOF.

Danksagung

Für die vielfältige Unterstützung mit Informationsmaterial und die Diskussionen von Teilergebnissen bedanke ich mich herzlich bei

- **Volker Leitzgen und Kollegen**, Microsoft Deutschland GmbH
- **Andre Ullmann**, Perseo Consulting Deutschland GmbH
- **Jakob Mayr**, exagon Consulting & Solutions GmbH
- **Holger Nickel und Kollegen**, ComConsult Kommunikationstechnik GmbH
- **Helmut Scherübl**, SMARTS GmbH
- **B. Carter Looney**, COC N-Tuition Business Solutions AG
- **Dennis Kirr und Kollegen**, Materna GmbH
- **Thomas Niedermeier und Kollegen**, Leibniz-Rechenzentrum

und bei den Betreuern Michael Brenner und Martin Sailer am Lehrstuhl.

Configuration Management, Beschreibung der Ausgangslage nach [Vog02]

'Es gibt wohl keinen Service Management Prozess, der schon mehr Verwirrung gestiftet hat, als das Configuration Management. In keinem anderen Bereich wurde bisher so viel Energie und Geld aufgewendet und gleichzeitig so wenig Nutzen erzielt. Unendlich viele Projekte zur Einführung dieses Prozesses und vor allem die dabei evaluierten Werkzeuge sind nach großem Aufwand im Sand stecken geblieben. Es scheint, dass jeder vom Ehrgeiz angesteckt ist, das Rätsel doch noch zu lösen.'

Inhaltsverzeichnis

1 Einführung

1.1 Motivation

Die wachsenden Anforderungen an IT-Abteilungen in Unternehmen haben diesen ehemals unterstützenden Geschäftsbereich zum integralen Bestandteil von Geschäftsmodellen gewandelt. Die Erfolgsabhängigkeit von einer funktionierenden IT-Abteilung führt im IT-Management zur verstärkten Einführung von qualitätssichernden Maßnahmen. Ein Qualitätsmerkmal ist eine IT-Umgebung, in der sich Abläufe möglichst einfach nachvollziehen lassen. Die Vereinfachung durch höhere Funktionsintegration und Automatisierung, die Verringerung der Typenvielfalt sowie Standardisierung der Infrastrukturkomponenten und die Verschlankung der IT-Infrastruktur sind Aufgaben zur Bewältigung einer zunehmenden Komplexität der vernetzten Systeme und aufsetzenden Businessprozesse. Vor dem Hintergrund einer anwachsenden IT-Integration in verschiedensten Geschäftsbereichen, zügig umzusetzenden Unternehmensstrategien und dem Bestreben nach einem schnelleren 'time-to-market', muss sich die IT möglichst flexibel den häufigen und schnellen Änderungen anpassen.
Um den Anforderungen aus diesen Entwicklungen nachzukommen, ist es erforderlich, das Wissen über die ständig in Veränderung befindliche IT-Infrastruktur erfolgreich zu managen. Denn mit diesem Wissen können Zusammenhänge zur Fehlervermeidung und Fehlerbehebung besser nachvollzogen sowie Optimierungspotenziale erkannt werden. Beispiele für die Bedeutung dieser Informationen zeigen IT-Mitarbeiterbefragungen in einer Studie von Remedy 'Managing Change in the Real World' [Fry04] auf. In einem der befragten Unternehmen beziehen sich nach nächtlichen Änderungsarbeiten in etwa 70 bis 80 Prozent der Incidents an Folgetagen auf diese Changes - deren Auswirkungen lassen sich offenbar nicht mehr ausreichend abschätzen und ziehen entsprechend Folgekosten nach sich. In einem anderen Unternehmen wird bemängelt, dass sich keiner mehr Überblick über die Abhängigkeiten innerhalb der IT-Infrastruktur verschaffen kann und diese Tatsache immer mehr zum Problem wird. Nicht nur wegen schlecht abschätzbaren Änderungsauswirkungen, sondern auch wegen Verzögerungen durch unbekannte Zuständigkeiten oder Fehlplanungen mangels Planungsgrößen zur IT-Infrastruktur.

1.2 Aufgabenstellung

Als Ansatz zur Lösung dieser Problematik schlagen die IT-Infrastructure Library (ITIL) und das Microsoft Operations Framework (MOF) 'Best Practices' für das Configuration Management vor. Das Configuration Management hat zur Aufgabe, ein logisches Modell der IT-Infrastruktur innerhalb einer Configuration Management Database (CMDB) bereitzustellen und zu verwalten. Dieses Modell schafft einen Überblick über die abhängigen Komponenten und dient anderen Service Management Prozessen als Informationsbasis. Fraglich ist allerdings, in welchem Umfang, mit welcher Strukturierung und welchem Verwaltungsaufwand dieses Modell gepflegt werden soll. ITIL und MOF geben dazu nur Ratschläge in der Art 'Erfassen Sie Abhängigkeiten innerhalb der CMDB', 'Identifizieren Sie Risiken' oder 'Wählen Sie zu erfassende Infrastrukturbestandteile bedacht aus'. Das ist allerdings ohne nähere Konkretisierung für eine Umsetzung zu wenig. Bei allen bekannten Herstellerlösungen für CMDBs ist auffällig, dass Datenstrukturen frei definierbar sind, hier also auch keine Vorgaben für eine zweckdienliche Infrastrukturerfassung gegeben werden. Offenbar werden diese Toollösungen von Anwendern jeweils mit unterschiedlicher Strukturierung verwendet, da gemeinsame, austauschbare Patterns fehlen oder aber auch am Bedarf vorbei gehen.
Diese Arbeit soll anhand eines Lösungsvorschlags ein Stück dazu beitragen, dass es einen Brückenschlag zwischen den ITIL/MOF Best Practices und der praktischen Einführung und Anwendung

von Configuration Management in Rechenzentren gibt. Die folgenden Fragestellungen werden dabei *unter anderem* behandelt:

Organisation des Configuration Managements: Lassen sich die Aufgaben des Configuration Managements nach ITIL und MOF, die in den beiden Frameworks gesammelt sind, in ein möglichst überschaubares und praxisnahes Prozessmodell zur Unterstützung der Umsetzung zusammenfassen?

Komplexitätsreduzierung der Datenerfassung: Nach welchen Kriterien lassen sich die in das Configuration Management einzubeziehenden IT-Komponenten der IT-Infrastruktur beschränken, um den Verwaltungsaufwand niedrig aber dennoch den Wert der Informationsbasis hoch zu halten? Können Abhängigkeiten zwischen den IT-Komponenten reduziert werden?

Datenmodell für eine CMDB: Durch welchen Ansatz wird Abhängigkeits- und Risikoanalyse möglichst einfach und vielfältig unterstützt? Wie müssen die Datenstrukturen gehalten sein, um häufigen Änderungen und individuellen Anpassungen nachkommen zu können?

Ausblick zur Entwicklung: Welche Rolle spielt zukünftig eine IT-Infrastrukturverwaltung gemäß ITIL/MOF-Configuration Management bei zunehmender Integration von IT in allen möglichen Bereichen des menschlichen Lebens?

Gleichzeitig sollte deutlich werden, dass die Einführung des Configuration Managements eine wichtige Basis für die Weiterentwicklung angrenzender und aufsetzender IT Service Management Prozesse schafft. Zusammenfassend dargestellt, befasst sich diese Arbeit mit der möglichst vollständigen Betrachtung der zentralen und kritischen Aspekte des Configuration Managements nach ITIL und MOF und zeigt, entlang des roten Fadens der *Strukturierung und Konkretisierung der ungenauen ITIL-/MOF-Empfehlungen*, umsetzungsorientierte Lösungsvorschläge auf.

1.3 Gliederung der Arbeit

In Kapitel 2 werden nach der Begriffsbildung die Service Management Prozesse der beiden Frameworks ITIL und MOF zusammengefasst. Die nachfolgenden Erläuterungen der Grundsätze ordnungsmäßiger Modellierung ermöglichen die Bewertung der Referenzmodell-Qualität von ITIL und MOF. Die Kenntnis des Umfangs, der Inhalte und der Qualität der beiden Frameworks ist in den Folgekapiteln Arbeitsgrundlage dafür, dass Modellvorgaben umfassend verwendet und für ungenaue ITIL- und MOF-Beschreibungen modellkonforme Festlegungen getroffen werden können - mit dem Ziel, einen adäquates Konzept für eine Umsetzung des Configuration Management darzulegen.

In Kapitel 3 werden in einem Anforderungskatalog die Anforderungen von ITIL und MOF an das Configuration Management gesammelt. Dieses Kapitel vermittelt an konkreten Modellbestandteilen, inwieweit ITIL und MOF Vorgaben für eine Umsetzung in einem Rechenzentrum geben und welche Fragestellungen offen bleiben. Darüber hinaus werden kritische Faktoren für die Umsetzung des Configuration Managements diskutiert, aus denen sich indirekt weitere Anforderungen ergeben.

Kapitel 4 enthält die Beschreibung einer Beispielinfrastruktur eines Rechenzentrums, die im Folgenden zur Verdeutlichung eines entwickelten Informationsmodells zur Datenmodellierung innerhalb einer CMDB (Kapitel 5) herangezogen wird.

In Kapitel 6 werden in einem entworfenen Arbeitsprozessmodell die Aktivitäten zum Configuration Management in einem zeitlichen und organisatorischen Rahmen strukturiert. Die beiden letztgenannten Kapitel enthalten mit der Präsentation eines Konzepts für eine Umsetzung den Schwerpunkt der Eigenentwicklungen innerhalb dieser Arbeit.

Abschließend erfolgt in Kapitel 7 die Zusammenfassung der vorhergehenden Kapitel sowie eine Einschätzung der zukünftigen Fortentwicklung des Configuration Managements.

Anmerkung zur Nennung von Quellen in der Ausarbeitung
Unter Berücksichtigung der Geschäftsinteressen der in der Danksagung genannten Firmen, wird teils bei Quellenangaben der unspezifische Vermerk [EXP] zu finden sein. Dieser weist auf eine Informationsquelle hin, die nach Vereinbarung nicht näher genannt wird. Diese Vereinfachung wird notwendig, nachdem das Thema dieser Arbeit bislang wissenschaftlich kaum aufgearbeitet wurde und damit nur beschränkt viele, öffentliche Informationsquellen zur Verfügung stehen, aus denen nachweisbar zitiert werden kann.

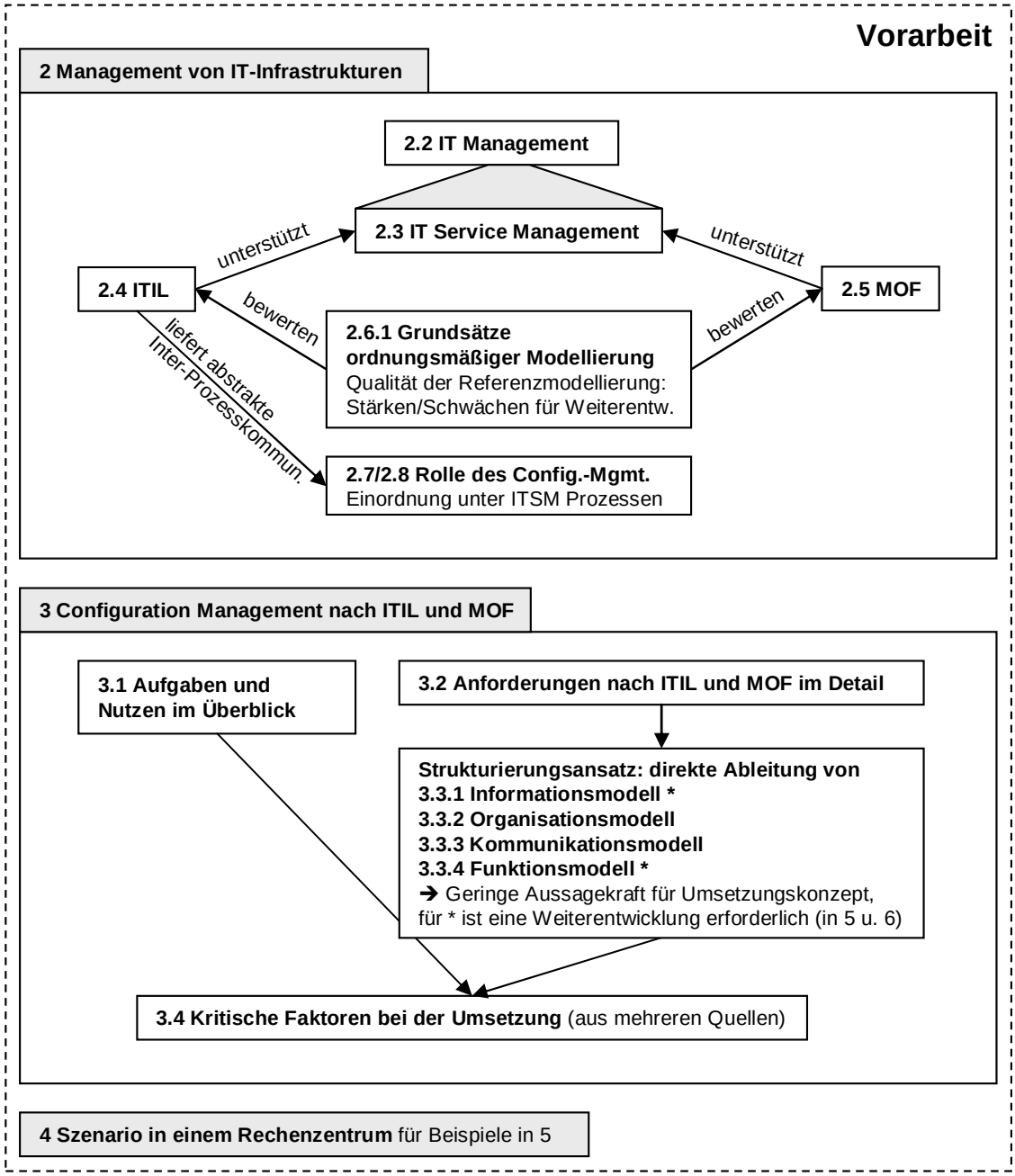


Abbildung 1-1: Vorgehensmodell 'Vorarbeit'

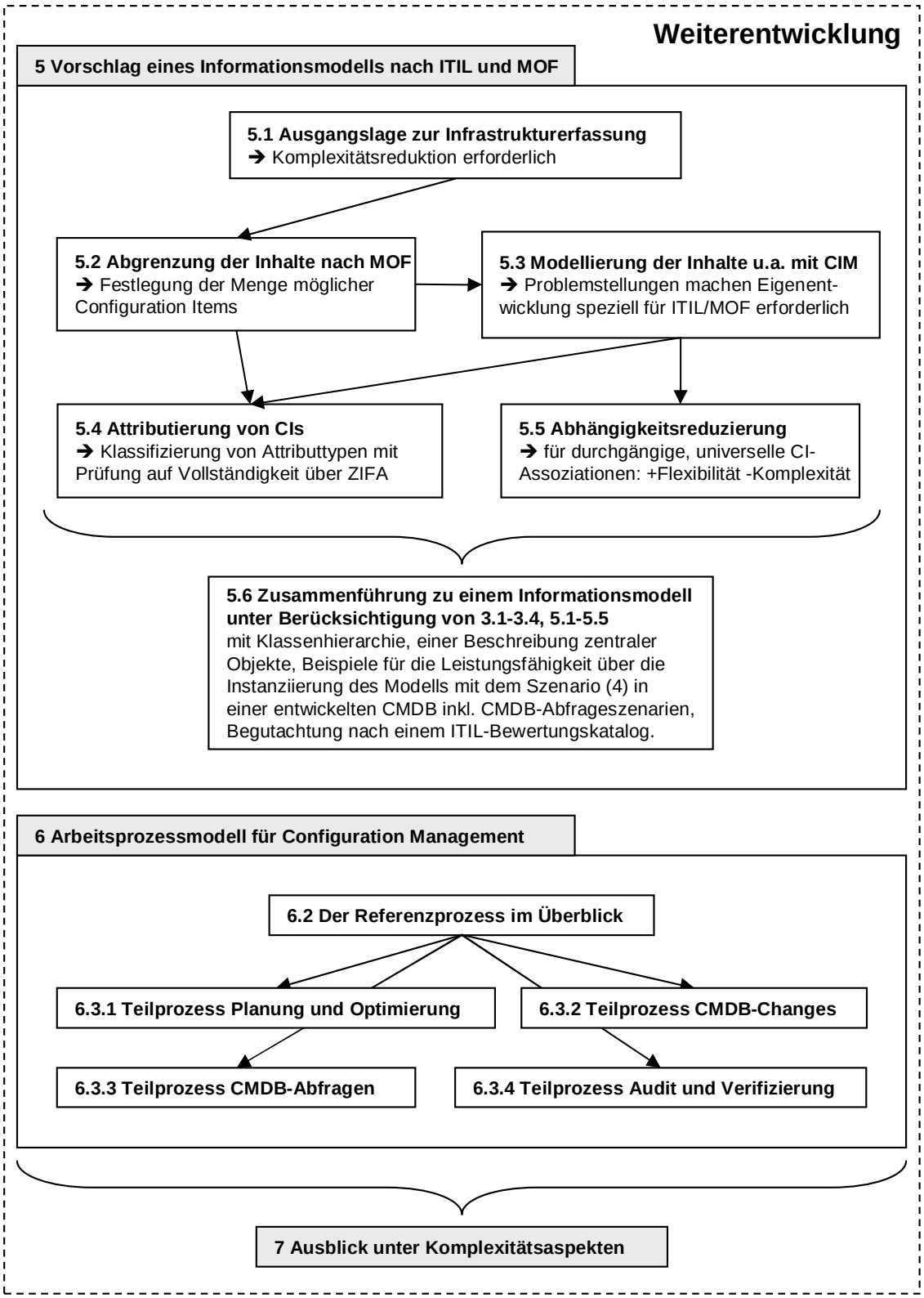


Abbildung 1-2: Vorgehensmodell 'Weiterentwicklung'

2 Management von IT-Infrastrukturen

2.1 IT-Infrastrukturen aus Managementsicht

Für Unternehmen ist das technische Management innerhalb der IT-Abteilung nur ein Teil der gesamten Managemenpyramide, die unter ihrer Spitze die 'Enabler' aus allen Geschäftsbereichen für ein erfolgreiches, strategisches Unternehmensmanagement untergliedert. Für jede Ebene der Pyramide gilt, dass eine übergeordnete Schicht nur dann effektiv gemanaged werden kann, wenn die untergeordneten Schichten effektiv arbeiten [HAN99]. Die in Abbildung 2-1 gezeigten, technischen Managementebenen lassen abschätzen, welche Bestandteile zu einer IT-Infrastruktur zu zählen sind. Die Zuordnung von Kunden als Abnehmer für IT-Dienstleistungen zur IT-Infrastruktur liegt im Sinne der Gesamtsicht einer IT, welche das Management der Technik der kundenorientierten Dienstbereitstellung unterordnet.

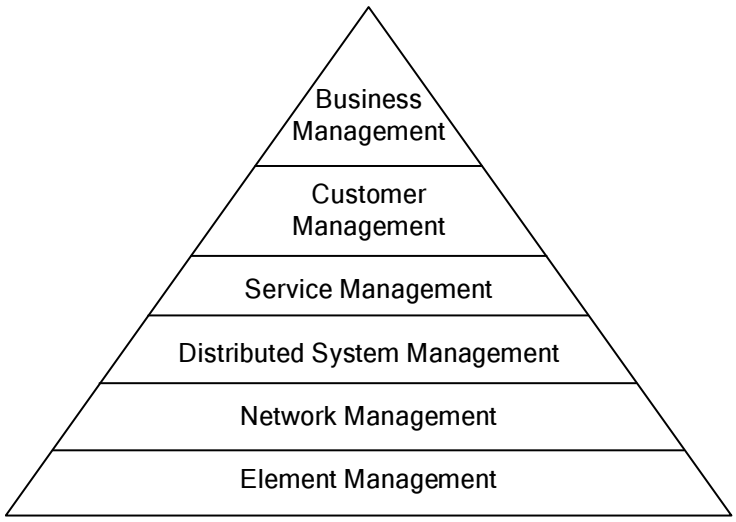


Abbildung 2-1: Managementpyramide nach [HAN99] in Anlehnung an ITU-T M.3010

An folgenden Beispielen sei verdeutlicht, wie die Managementsicht für IT-Dienstleister die Abgrenzung dessen bestimmt, was zu ihrer IT-Infrastruktur zu zählen ist. Ein IT-Service Provider, der in einem Rechenzentrum Serverhousing anbietet, beschäftigt sich mit der Administration von Netzen und Serverhardware. Letztere zählt er zu seiner IT-Infrastruktur, während die Anwendungen seiner Kunden, die selbsttätig Applikationsverwaltung auf der bereitgestellten Serverhardware betreiben, nicht dazu gezählt werden. Ein Reseller eines Dienstes, wie beispielsweise einer Kooperationsplattform mit Filesharing, Projektverwaltung und gemeinsamen Emailfoldern, wird diese Dienstbestandteile als seine IT-Infrastruktur betrachten, nicht aber das zugrunde liegende Betriebssystem, die Rechensysteme und Netze. Full Service Provider haben den Anspruch, für Kunden End-to-End-Prozesse abzubilden und sollten insofern alle Prozessbestandteile zu ihrer IT-Infrastruktur zählen.
Dieser Vergleich zeigt die Unterschiede im Verständnis einer IT-Infrastruktur und legt gleichzeitig nahe, dass für die technische Leistungsbereitstellung jeweils Management aus zusammen liegenden, unteren Schichten der Managementpyramide betrieben wird, während die oberen, nichttechnischen Ebenen ab dem Service Management gemeinhin gleichartig benötigt werden. Bevor

nun der Begriff des Service Managements erläutert wird, werden zunächst Grundlagen des IT Managements im Überblick eingeführt, auf die im späteren Verlauf Bezug genommen wird.

2.2 IT Management

Allgemein beschrieben, umfasst das Management vernetzter Systeme alle Maßnahmen, die einen effektiven und effizienten, an den Zielen des Unternehmens ausgerichteten Betrieb der Systeme und ihrer Ressourcen sicherstellen [HAN99]. Effektives Management bedeutet, mit allen möglichen Mitteln das Ziel des Managements zu erreichen. Als effizentes Management wird die Verwendung der zur Verfügung stehenden Mittel zur Erreichung des maximal möglichen Ergebnisses bezeichnet.
Die Strukturierung des Managements über eine Systematik hilft dabei, Ressourcen, Organisationen, Aufgaben, Tools und Modelle, die in das Management einbezogen sind, überschaubar und vergleichbar zu machen. Als Beispiel für eine Systematik seien die Prinzipien des OSI-Managements als Vorschlag der ISO genannt, das folgende Aspekte des Managements unterteilt:

Funktionsmodell zur Strukturierung der Managementfunktionalität nach den Funktionsbereichen **F**ault, **C**onfiguration, **A**ccounting, **P**erformance und **S**ecurity Management (FCAPS). Diese Funktionsbereiche klassifizieren möglichst vollständig alle Funktionen auf (technische) Objekte des Managements, orthogonal zu deren Ebeneneinordnung (vgl. Managementobjekte in 2-1). Das bedeutet beispielsweise, dass die Funktion des Konfigurationsmanagements sowohl auf einzelne Elemente angewandt wird, als auch auf zusammenfassende Netze und verteilte Systeme. Vorgreifend auf die Einführung von ITIL und MOF als Rahmenwerke für das Service Management sei darauf hingewiesen, dass die Funktionsbereiche des OSI-Managements eine grobe Zuordnung zu ITIL und MOF Funktionen zulassen. Dies kann in der Verallgemeinerung, ohne speziellen Bezug zu ITIL und MOF, als Bestätigung der ebenenübergreifenden, möglichst vollständigen Klassifikation des Managements nach FCAPS betrachtet werden.

Organisationsmodell zur Behandlung und Unterstützung von Organisationsaspekten, Rollen und Kooperationsformen eines verteilten, kooperativen Managements in einem Netz offener Systeme. Im technischen Management wird dabei insbesondere die Organisation von Managementsystemen und davon überwachten beziehungsweise gesteuerten Managementobjekten beschrieben.

Kommunikationsmodell zur Beschreibung der Kommunikationsvorgänge zum Austausch von Managementinformationen. Beispielsweise sind im technischen Management Spezifikationen von Protokollen für den Datenaustausch Teil des Kommunikationsmodells.

Informationsmodell zur Beschreibung der IT-Infrastruktur über Managementobjekte, welche die Bestandteile einer sogenannten Management Information Base (MIB) darstellen. Im weiteren Verlauf wird auf die Prinzipien eines Informationsmodells, als Basis für das Configuration Management, beziehungsweise für die ITIL und MOF Funktionen im Allgemeinen, insbesondere eingegangen werden.

Managementobjekte (Managed Objects) sind nach dem Vorschlag der ISO objektorientiert modelliert und enthalten im Wesentlichen folgende Elemente:

- Attribute des Managed Objects, die nach außen sichtbar sind
- zulässige Operationen auf das Managed Object
- Notifications aus dem Managed Object
- Verhaltensbeschreibung zur reduzierten, funktionalen Betrachtung des Managed Objects.

Managed Objects weisen verschiedene Relationen auf, beispielsweise die Einordnung in einen Vererbungsbaum zur Übernahme von Elementen der Parentklassen oder die Einordnung in einen

Enthaltenseinsbaum, der für eine Komponente die Subkomponenten zusammenfasst.

Unabhängig von der detaillierten Beschreibung des OSI-Managements, sei folgendes Beispiel zur Verdeutlichung der praktischen Bedeutung der Modellierungsgrundlagen für Managed Objects genannt. Über eine MIB als Informationsbasis über vorhandene, Managed Objects, lassen sich per SNMP-Protokoll die 'SysContacts' und 'SysLocations' beliebiger Netzwerkkomponenten, sofern gesetzt und freigegeben, auslesen. Des Weiteren können beispielsweise entsprechende Filter gesetzt werden, so dass nur diejenigen Netzwerkkomponenten gezeigt werden, die über ein bestimmtes Gateway mit anderen Netzen kommunizieren. Zwar ist der Detaillierungsgrad in diesem Beispiel noch nicht allzu groß, doch lässt sich daran vermuten, dass die Attribute des Komponentenmanagements vielfältig ausfallen können und gerade bei einer nicht automatisierten Erfassung eine Selektion der 'wesentlichen' Informationen unabdingbar ist. Die **Problematik des Managements** ist, dass einerseits bei Erfassung von zu vielen Informationen der Pflegeaufwand für Managementinformationen nicht mit dem Erfolg aus den Managementtätigkeiten zu rechtfertigen ist. Letzteres verstößt gegen den Effizienzanspruch des Managements. Bei Erfassung von zu wenigen Informationen kann das Management mangels Aussagekraft der Informationsbasis allerdings keine adäquate Steuerung übernehmen, womit der Effektivitätsanspruch des Managements verletzt ist.

Für ein effizienteres Management ist folglich eine fortschreitende Automatisierung und Standardisierung bei der Erfassung und Verwaltung von Infrastrukturkomponenten wünschenswert. Eine häufige Problematik bei der Entwicklung von Komponenten ist, dass die Bereitstellung der eigentlichen Funktionalität vorrangig vor Managementaspekten ist. Der Bedarf an Funktionen für ein Monitoring und das Steuern einer Komponente ist oftmals bei Inbetriebnahme zweitrangig. Seine Bedeutung wird erst mit den Erfahrungen aus dem Betrieb erkannt. Während auf Ebene der Netzwerkkomponenten mit SNMP ein etablierter Managementstandard vorhanden ist, zeigen sich gerade in der Softwareentwicklung auf Applikationsebene Defizite bezüglich durchgesetzter Standards. So fehlen im Allgemeinen Standardschnittstellen zum Test der Erreichbarkeit, zum Auslesen der Konfiguration und zur Messung der Ressourcennutzung. Die zunehmende Vernetzung von Applikationen, als Stichwort dafür sei Enterprise Application Integration (EAI) genannt, schafft somit Abhängigkeitsnetze, die nur mehr uneinheitlich gemanaged werden können. Das verkompliziert das in der Management-Pyramide auf Applikationen, Systeme, Netze usw. aufsetzende IT Service Management.

2.3 IT Service Management

IT Service Management (*ITSM*) bezeichnet die Gesamtheit der Maßnahmen, um die bestmögliche Unterstützung von Geschäftsprozessen (*GP*) durch die IT-Organisation zu ermöglichen [Som04]. Ein *Service* ist eine Sammlung von physischen und logischen (IT- und nicht IT-) Komponenten, welche den jeweiligen Geschäftsprozessen zugeordnet sind [PQ03], mit dem Ziel, deren Unterstützung sicher zu stellen. Der Begriff *IT Service* bezeichnet die Menge aller Funktionalitäten, die durch einen Diensterbringer an einer Dienstschnittstelle dem Dienstnehmer zur Verfügung gestellt werden [Som04]. Laut [Kai99] gibt es für den Service-Begriff in verschiedenen Standards keine einheitliche Definition, insofern wurden oben, als Vorgriff auf spätere Ausführungen, die Begriffserklärungen des Microsoft Operations Frameworks (MOF) gewählt. Die Grundlage für die Erbringung von Services bilden Service Level Agreements (SLAs), die zwischen Kunden und IT-Organisationen vereinbart werden. Operative Tätigkeiten sind dem Service Management untergeordnet.
In Abbildung 2-2 werden verschiedene, einem Service zugeordnete IT-Komponenten beispielhaft dargestellt.

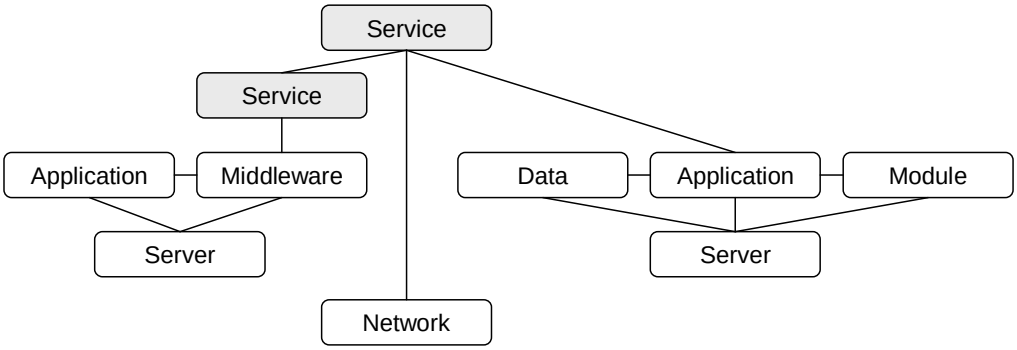


Abbildung 2-2: Beispielkomponenten eines Servicegraphen

2.4 Die ITIL zur Unterstützung des IT Service Managements

Die IT Infrastructure Library (ITIL) zur Unterstützung des ITSM über die Beschreibung von betreiberorientierten IT Service Management Prozessen, wird in diesem Kapitel in einer Zusammenfassung vorgestellt, um die Verknüpfungen mit und die Anforderungen an den Prozess des Configuration Managements zu vermitteln.

Die ITIL wurde Ende der 80er Jahre als 'Best Practice' Ansatz durch das britische Office of Government Commerce (OGC) entwickelt. Mittlerweile haben weit über 10.000 Unternehmen das ITIL-Framework angenommen und es gibt weltweit über 50.000 zertifizierte ITIL-Professionals [Som04]. Die ITIL hat in den letzten Jahren mit ihrer starken Verbreitung in IT-Abteilungen dazu beigetragen, dass sich unter IT-Verantwortlichen ein Bewusstsein und eine gemeinsame Terminologie für IT Service Management herausgebildet haben [Mat04]. Das gemeinsame Verständnis erleichtert die Kommunikation zwischen IT-Abteilungen, sowohl innerhalb eines Unternehmens als auch unternehmensübergreifend. Die ITIL wird als defacto Standard durch das IT Service Management Forum (itSMF) gefördert und weiter entwickelt. Der Vollständigkeit wegen sei darauf hingewiesen, dass die ITIL nicht nur IT Service Management beschreibt, sondern beispielsweise auch administrative Tätigkeiten oder 'Businessperspektiven'.

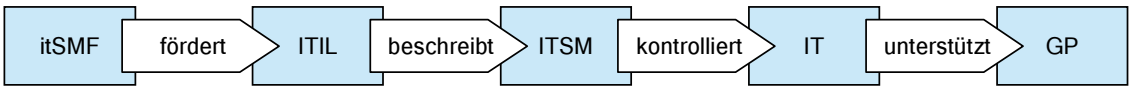


Abbildung 2-3: Wirkungskette vom itSMF bis zu Geschäftsprozessen

2.4.1 Vorstellung der ITIL-Prozesse

Das in Anlehnung an das IPW-Modell[1] für das Configuration Management durch die ComConsult GmbH erweiterte Modell in Abbildung 2-4 [Jak04], setzt die ITIL-Prozesse wie auch angrenzende Funktionen und Rollen einer IT-Infrastruktur über Informationsflüsse für einen Gesamtüberblick in Verbindung. Die ITIL beschreibt zehn zentrale ITSM Prozesse aus den Bereichen Service Support und Delivery, die in der Abbildung mit einem (X) gekennzeichnet sind. Die folgenden Beschreibungen wurden in der Reihenfolge der Abbildung aufgenommen und basieren unter anderem auf dem ITIL-Syllabus der Firma dv-werk GmbH [DKW04].

[1] Das IPW-Modell (Implementation of Process-Oriented Workflow) wurde 1991 von dem niederländischen Beratungsunternehmen Quint Wellington Redwood in Zusammenarbeit mit der niederländischen PTT Telekom entwickelt. Es basiert auf der ITIL und erweitert diese bezüglich weiterer strategischer, taktischer und operationaler Bestandteile einer IT-Organisation [Mag97].

Allgemeines zur ITIL-Terminologie

Service Delivery [OGC01] Die untergeordneten Prozesse und Funktionen befassen sich mit der langfristigen Planung und Optimierung der IT-Dienstleistungen. Gemäß Abbildung 2-4 fallen darunter das Account-Management, also die IT-eigene Leistungsüberwachung, die Service Planung für ITSM Prozesse (Financial-, Capacity-, Availability- und Continuity Management) sowie die Service-Entwicklung [OGC02b] und das Security Management.

Service Support [OGC00] Die untergeordneten Prozesse und Funktionen unterstützen die operative Erbringung von IT-Services. Darunter fallen das Incident-, Problem-, Change-, Release- und Configuration Management sowie das Application und ICT Infrastructure Management. Diese Begriffe werden weiter unten genauer erläutert.

Kunde Die ITIL geht von einer zweiteiligen Sicht auf den Kunden aus. Zum einen gibt es den Kunden (customer) als Vertragspartner, der Leistungen (IT-Services) und Gegenleistungen (Bezahlung/Kostenträgerschaft) vereinbart, die Leistungserfüllung überwacht und innerhalb seiner eigenen Organisation die Nutzung der Leistungen rechtfertigen muss. Zum anderen gibt es den Benutzer (user), der die Leistungen im Rahmen seiner operativen Tätigkeiten nutzt.

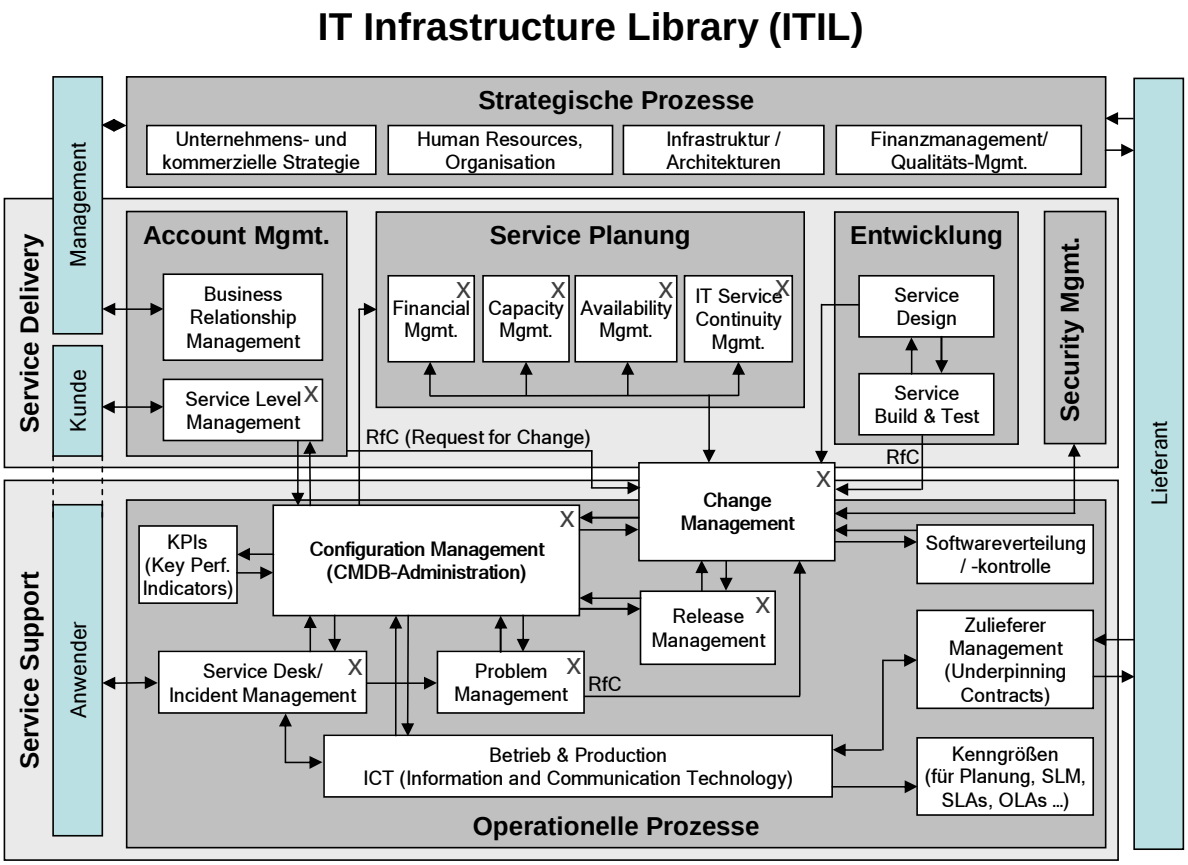


Abbildung 2-4: Übersicht über ITIL-Prozesse, ComConsult GmbH nach IPW™ Modell

Im Folgenden werden nun die einzelnen Bestandteile der Abbildung 2-4 in den Bereichen Service Support und Delivery gemäß ihrer Reihenfolge erläutert. Für weiterführende Informationen sind großteils in geschweiften Klammern die Titel der entsprechenden ITIL-Bücher des OGC genannt.

Account Management

Business Relationship Management ***{The Business Perspective}*** Kernziel ist es, eine gemeinsame Kommunikationsbasis zwischen dem Kunden (Business) und dem IT-Service Anbieter (Provider) zu finden. Dem Kunden soll ein Verständnis für IT-Services vermittelt werden, deren Nutzung im Accounting festgehalten wird.

Service Level Management ***{Service Delivery}*** Die Aufgabe des Service Level Managements ist die Anpassung der Services an die Wünsche und Anforderungen des Kunden und das Aushandeln von Dienstleistungsvereinbarungen (Service Level Agreements - SLAs) inklusive Überwachung der Einhaltung vereinbarter Service Level. Dabei wird mit dem Kunden über Ende-zu-Ende-Services verhandelt, die interne, technische Umsetzung wird ausgeblendet.

Service Planung

Financial Management ***{Service Delivery}*** Übernimmt die Kosten- und Leistungsverrechnung, die Budgetierung (Vorhersage und Kontrolle der Ausgaben) und stellt dem Vertragsmanagement wie auch dem Finanzmanagement des Unternehmens Informationen über Ausgaben für angefragte IT-Services zur Verfügung.

Capacity Management ***{Service Delivery}*** Befasst sich mit der langfristigen Planung von Kapazitäten für die Erbringung von IT-Services wie menschliche oder technologische Ressourcen mit Betrachtung zeitlicher Bereitstellungsaspekte.

Availability Management ***{Service Delivery}*** Managed die Hilfsmittel zur Vermeidung oder möglichst schnellen Behebung von Ausfällen, beispielsweise durch die Forderung von Monitoring und Backups. Ziel ist die Steigerung der Zuverlässigkeit der Services.

IT Service Continuity Management ***{Service Delivery}*** Dient zur Erarbeitung von Maßnahmen, die zur Weiterführung der Dienste nach Totalausfällen von Teilen der IT-Infrastruktur zu ergreifen sind.

Service-Entwicklung

Service Design ***{Planning to Implement Service Management}*** Sammeln von Anforderungen, Ist-Zustand und Zieldefinition zum Design eines IT-Services. Dabei werden die Ergebnisse aus den Prozessen der Service Planung angewandt.

Service Build and Test ***{Planning to Implement Service Management}*** Implementierung, Test und Einführung eines IT-Services in Koordination mit dem Change Management.

Security Management

{Security Management} Hat das Ziel, Sicherheitsanforderungen aus SLAs, sonstigen Verträgen, Gesetzen und Sicherheitspolicies des Unternehmens zu erfüllen. Es schlägt Maßnahmen und Richtlinien vor, um im Operating diesen Sicherheitsanforderungen nachzukommen.

Operationelle Prozesse

Service Desk/Incident Management ***{Service Support}*** Das Incident Management nimmt von Anwendern oder aus der IT-Administration Vorfälle wie Störungsinformationen oder Fragen zur IT an, klassifiziert diese und versucht, Lösungen bereitzustellen.

Problem Management ***{Service Support}*** Werden keine Lösungen im Incident Management gefunden, werden Anfragen an das Problem Management eskaliert, das sich mit der Ursachenforschung für Probleme beschäftigt. Bei notwendigen Änderungen an der IT-Infrastruktur werden Requests for Changes (RFCs) an das Change Management überstellt. Aufgabe ist auch, potentielle Störungen durch präventive Maßnahmen zu begrenzen.

Change Management ***{Service Support}*** Das Change Management koordiniert die Durchführung von angetragenen Changes (RFCs) und überwacht dabei die betroffenen Bestandteile der IT über Abfragen der CMDB (Configuration Management Database). Es ist damit in hohem Maße von der Bereitstellung genauer Konfigurationsdaten durch das Configuration Management abhängig.

Release Management ***{Service Support}*** Das Release Management ist dem Change-Management beigeordnet und koordiniert die Einführung (Roll-Out) autorisierter Changes in die IT-Infrastruktur. Häufig werden lediglich größere, zusammengehörige Change-Aktionen als Release bezeichnet.

Configuration Management ***{Service Support}*** Das Configuration Management stellt in einer CMDB (Configuration Management Database) eine Informationsbasis über die in Services einbezogenen Komponenten der IT-Infrastruktur, abgebildet in Configuration Items (CIs), bereit. Es überwacht den Status der erfassten CIs und ist für die Aktualität der Daten in der CMDB verantwortlich. Unter den Prozessen der Service Delivery und Support hat das Configuration Management wegen des vielseitigen Bedarfs an Konfigurationsinformationen eine zentrale Bedeutung, beispielsweise zur Erfassung von Abhängigkeiten zu Incidents, Problems, Changes und Releases.

Softwareverteilung/-kontrolle ***{Software Asset Management, teils Application Management}*** Überwacht im Wesentlichen den Einsatz gültiger Lizenzen und steht für die Softwareverteilung in enger Verbindung mit dem Change- und Release Management.

ICT Infrastructure Management ***{ICT Infrastructure Management}*** Befasst sich mit der Beschreibung administrativer Tätigkeiten zur (täglichen) Erbringung von IT-Services, beispielsweise Durchführung von Backups, IT-Monitoring, Bereitstellen von Speicherplatz usw. Es besteht wegen des Ineinandergreifens von Planung/Optimierung und Durchführung eine enge Verbindung zu allen oben genannten Prozessen.

2.4.2 Monitor-Control-Loop als generelles Prinzip in der ITIL

Die ITIL lebt durch die ständige Optimierung der Prozessbeschreibungen - selbiges gilt für die individuell umgesetzten Prozesse in den IT-Abteilungen. Der Monitor-Control-Loop zeigt die Schritte zur Überwachung und Anpassung der entsprechenden Prozessbestandteile. Dieses Prinzip entspricht letztendlich dem häufig zitierten Deming-Cycle ('Plan-Do-Check-Act'), der sich auch im MOF-Prozessmodell wieder findet. Auf dieses Schaubild wird später noch bei der Behandlung der Optimierung von Configuration Management, beziehungsweise der Optimierung der zu erfassenden IT-Infrastruktur in der CMDB, hingewiesen werden.

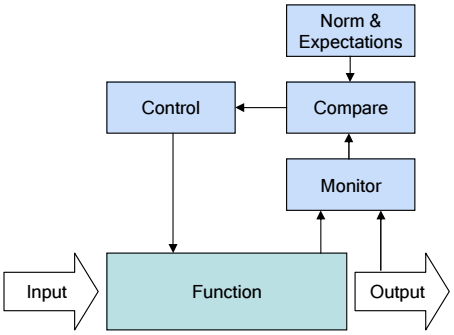


Abbildung 2-5: Monitor-Control-Loop aus [OGC02a]

2.5 Das MOF im Überblick

Nachdem nun die ITIL in Kürze mit ihren Prozessen und Funktionen beschrieben wurde, soll nun das zweite Framework vorgestellt werden, das für die Anforderungserhebung in Bezug auf Configuration Management innerhalb dieser Arbeit herangezogen wurde. Das Microsoft Operations Framework (MOF) erschien im Jahr 2000 als Anpassung und Weiterentwicklung der ITIL in Bezug auf Microsoft Technologien [Som04] und liegt inzwischen in der dritten Fassung vor. In großen Teilen ist dieses Framework dennoch technologieunabhängig und steht im Internet frei zum Download zur Verfügung [Mic04e].

2.5.1 Die MOF-Modelle

MOF besteht aus drei grundlegenden Modellen, die jeweils eine 'Hauptkomponente des IT-Betriebs' repräsentieren [PQ03].

Das **Prozessmodell** stellt in einem Optimierungszyklus Service Management Functions (SMF, entsprechend dem Prozess-Begriff in der ITIL-Terminologie - beide Bezeichnungen werden im Folgenden äquivalent verwendet) dar, die von Service-Organisationen verwendet werden, um IT-Services zu erbringen, zu verwalten und zu warten. Es erweitert die ITIL-Prozesse um weitere Prozessdefinitionen, insbesondere im Bereich des Operatings. Im nächsten Kapitel wird das Prozessmodell näher betrachtet.

Das **Teammodell** beschreibt die nötigen Rollen für den IT-Betrieb und unterstützt das Management bei der Personalorganisation. Die Rollen können jeweils einem oder mehreren Prozessen zugeordnet werden. So gibt es beispielsweise für den Prozess des 'Storage Managements' die Rolle des 'Storage Managers', die zum Team des 'Betriebs-Rollenclusters' gehört. Nachdem die Nutzung des Configuration Managements beziehungsweise gewisser Configuration Items den im Teammodell beschriebenen Rollen im Allgemeinen nicht eindeutig genug zugeordnet werden kann, wird an dieser Stelle vorweg gegriffen, dass in den späteren Kapiteln dem Teammodell keine größere Bedeutung zukommen wird.

Das **Risikomodell** (oder ab MOF 3: Risc Management Discipline) dient im alltäglichen Management zur Einschätzung von Betriebsrisiken über die Festlegung von Betriebsprinzipien und die Anwendung eines fünfstufigen Bewertungsprozesses. Ein Risiko ist das Produkt aus Ausfallwahrscheinlichkeit und finanzieller Auswirkung. Beide Größen müssen im Rahmen der administrativen Tätigkeiten ständig geprüft werden, um gegebenenfalls korrigierende Maßnahmen einzuleiten. Für ihre Abschätzung hält die CMDB aus dem Configuration Management wichtige Daten bereit, wie beispielsweise Abhängigkeitsinformationen und möglicherweise auch Ausfallwahrscheinlichkeiten von Teilkomponenten. Beim Design des vorgeschlagenen Informationsmodells für eine CMDB in Kapitel 5 wurden die Belange des Risikomanagements mit berücksichtigt.

2.5.2 Das erweiterte MOF-Prozessmodell

Im erweiterten Prozessmodell sind alle zehn zentralen ITSM Prozesse sowie das Infrastructure und Security Management aus der ITIL enthalten. Die in der Abbildung 2-6 kursiv dargestellten Service Management Functions wurden im MOF hinzugenommen. Die Prozesse innerhalb der Quadranten des Optimierungskreislaufes finden gleichzeitig statt - es gibt also keine phasenversetzte Abarbeitung der Prozesse, wie die Darstellung möglicherweise suggeriert. Stattdessen soll vermittelt werden, dass die durch Prozesse gemanagten Services einem Lifecycle beziehungsweise dem oben genannten Monitor-Control-Loop-Prinzip unterliegen. Innerhalb dieses Optimierungszyklus wird auf die entsprechenden ITSM Prozesse zurückgegriffen und aufgrund ihrer Ergebnisse zusammenfassende Reviews des betrachteten Services erstellt.

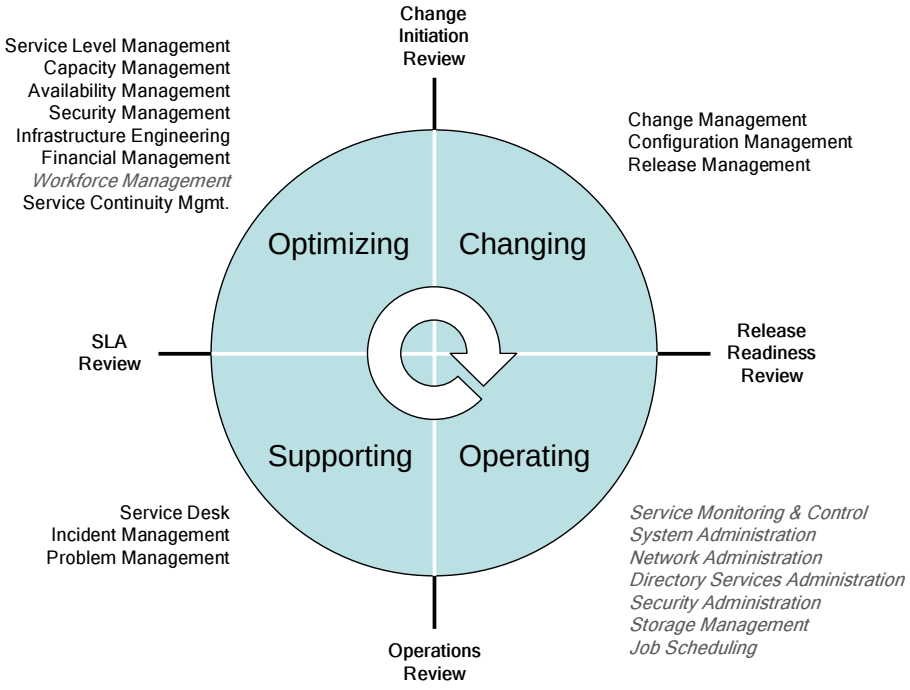


Abbildung 2-6: Prozessmodell nach MOF

Die im MOF hinzugefügten Prozesse haben folgende Aufgaben:

Workforce Management Stellt sicher, dass geeignetes Personal für die Unterstützung der Prozesse bereitgestellt wird. Dafür beschäftigt es sich mit Recruiting, Entwicklung und Erhaltung der IT-Mitarbeiter.

System Administration Ausübung und Steuerung der täglichen Betriebs- und Administrationsaufgaben für Systeme. Hierzu gehören beispielsweise Netzwerkverwaltung, Administration von File-, Print- und Anwendungsservern, Benutzerverwaltung und die Bereitstellung von Überwachungsfunktionen.

Security Administration Stellt sicher, dass Daten innerhalb der Betriebsumgebung abgesichert sind. Dazu gehört die Gewährleistung der Vertraulichkeit und Integrität der Daten mit unterschiedlichen Berechtigungsstufen.

Service Monitoring and Control Aufgabe ist die Überwachung und Event-Behandlung innerhalb der IT-Infrastruktur inklusive Eventkorrelation zur gezielten Weitergabe von Störungen. Events werden durch Regeln ausgelöst, die beispielsweise über SLAs vereinbart wurden.

Job Scheduling Stellt sicher, dass die verfügbare Rechnerkapazität effektiv und gleichmäßig genutzt wird, unter Berücksichtigung vereinbarter Service-Levels. Dies ist insbesondere für automatisierte Routineaufgaben relevant (beispielsweise Backupzeiten, Zeiten für Logfile-Rotationen und -Auswertungen).

Network Administration Stellt den Prozess zum Betrieb und zur Wartung der physischen Netzwerkkomponenten dar. Das Ziel besteht darin, die Verfügbarkeit der Netzwerkverbindungen und die allgemeine Performance sicherzustellen.

Directory-Services-Administration Verwaltung eines Directory-Dienstes für vereinfachte Authentifizierung und Auskünfte im Netzwerk; Bestimmung von Objektprofilen und Ändern von Objekten.

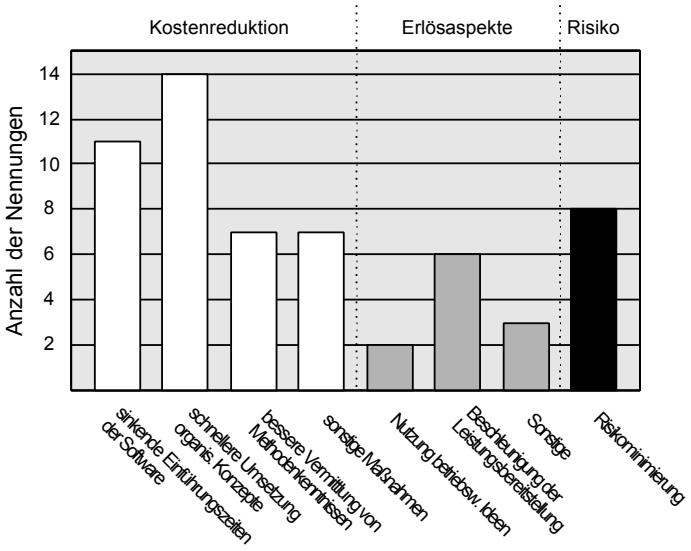


Abbildung 2-7: Mit Referenzmodellen verfolgte Zielsetzungen [BSIG00]

Storage Management Beschäftigt sich mit der Speicherung, der Wiederherstellung und Archivierung von Daten und entwickelt für diese Tätigkeiten Pläne und Richtlinien.

2.6 Bewertung von ITIL und MOF für die Referenzmodellierung

Nachdem nun die beiden Frameworks im Überblick vorgestellt wurden, soll nun im folgenden Kapitel die Qualität von ITIL und MOF für die Referenzmodellierung bewertet werden. Die Erkenntnisse aus der Bewertung sind später der Konzeptentwicklung für Configuration Management zugrunde zu legen, das heißt, mit dem *Wissen über Stärken und Schwächen der beiden Frameworks* können später begründet Festlegungen und Erweiterungen vorgenommen werden.

Referenzmodellierung kann umschrieben werden als die Menge aller Handlungen, welche die Konstruktion und Anwendung wiederverwendbarer Modelle (Referenzmodelle) beabsichtigen [FL04]. Gemäß einer Befragung unter Referenzmodellnutzern [BSIG00] werden Referenzmodelle zur Orientierung in der betrieblichen Praxis mit den Zielsetzungen aus Abbildung 2-7 herangezogen. Übergreifende Ziele sind dabei die Risikominimierung und die Beschleunigung bei der Einführung betrieblicher Informationssysteme für einen besseren ROI.

In der Referenzmodellierungsforschung wurden von Schütte mit den Grundsätzen ordnungsmäßiger Modellierung (GoM) I [BRS95] und II [Sch98] Kriterien zur Einordnung betrieblicher Informationsmodelle entwickelt. Sie dienen insbesondere dazu, die Qualität von Referenzmodellen zu bewerten. Hochstein verwendet die GoM II in [HZB04] zur formalen Bewertung des ITIL-Referenzmodells. Seine Ergebnisse sind in Tabelle 2-8 zusammengefasst. Für ein besseres Verständnis der Tabelle werden nun die Grundsätze der GoM II erläutert. Nachfolgend wird eine vergleichbare Anwendung der GoM auf das MOF vorgenommen, um Schlussfolgerungen für die Konzeptentwicklung für Configuration Management innerhalb dieser Arbeit auf der Grundlage beider Frameworks ziehen zu können.

2.6.1 Grundsätze ordnungsmäßiger Modellierung (GoM)

Die folgenden Beschreibungen basieren auf [BSIG00] und [Mei00].

Grundsatz der Konstruktionsadäquanz Ein Modellnutzer orientiert sich an einem Referenzmodell, um ein definiertes Problem schneller und zielgerichtet zu lösen. Der *Grundsatz der Konstruktionsadäquanz* fordert die 'problemangemessene Nachvollziehbarkeit der Modellkonstruktion'. Zunächst muss über das Problem sowie die Sichtweise und den Umfang des zu modellierenden Umweltausschnitts Einigung erzielt werden. Die Festlegung der relevanten Systembestandteile und abbildender Informationsobjekte ist eine Konstruktionsleistung des Modellbildenden und damit Teil seiner Wahrnehmung. Der Abstraktionsgrad bestimmt dabei im Wesentlichen die spätere Nutzbarkeit des Modells aus Anwendersicht. Für Informationsobjekte wird Minimalität verlangt, so dass kein Informationsobjekt ohne Informationsverlust für das Modell entfernt werden kann. Festzulegen bleibt, was modellspezifisch Informationsverlust erzeugt.
Des Weiteren ist eine Einigung bezüglich der Modelldarstellung zu erreichen. Erstrebenswert ist zum einen Intra-Modellkonsistenz, um innerhalb eines Modells wiederkehrende Konstrukte gleichartig zu modellieren. Zum anderen soll Inter-Modellkonsistenz erreicht werden, die besagt, dass reale Sachverhalte in unterschiedlichen Modellen einheitlich dargestellt werden, um auf vorhandene Problemlösungen zurück greifen zu können.
Anhand der praktischen Nutzbarkeit des Modells kann im Nachhinein dessen Qualität beurteilt werden.

Grundsatz der Sprachadäquanz Bei der Untersuchung der verwendeten Sprache für das Modellsystem wird zwischen Spracheignung und Sprachrichtigkeit unterschieden. Zur Spracheignung gehört eine ausreichende, semantische Mächtigkeit und - abhängig vom Modellzweck - ein gewisser Formalisierungsgrad der Sprache (für die Umsetzung einer automatischen Prozessteuerung bietet sich beispielsweise ein hoher Formalisierungsgrad an). Unter dem Kriterium der Sprachrichtigkeit wird die Konsistenz und Vollständigkeit des Modells gegenüber dem Metamodell beurteilt. Die Sprachverständlichkeit wird durch die Kenntnisse des Anwenders bestimmt.

Grundsatz der Wirtschaftlichkeit Die Verwendung eines Referenzmodells erzeugt einerseits Kosten im Zusammenhang mit der Erstellung und Einführung, andererseits sollten Kostenvorteile durch das Aufgreifen bestehender Modelle entstehen. Der Kostenfaktor verhält sich meist konträr zur Umsetzung der anderen Grundsätze.
Die Wirtschaftlichkeit wird des Weiteren durch die Modellrobustheit bestimmt. Ein Modell ist robust, wenn die Modellbausteine gegenüber Veränderungen gültig bleiben. Weiterhin wird Modellflexibilität gefordert, welche die Anwendbarkeit von Änderungen am Modell bestimmt. Die Sprachadäquanz unter wirtschaftlichen Gesichtspunkten bestimmt die am Modellzweck festgemachte Anwendbarkeit der Modellierungssprache. Dient ein Modell zur Vermittlung gegenüber fachlich ungeschulten Ansprechpartnern, so ist eine informelle Sprache vorteilhafter. Soll auf einem Modell eine Simulation angewendet werden, ist eine formale Beschreibung wirtschaftlicher. Die Übersetzbarkeit der gewählten Sprache ist ein weiteres Kriterium, das je nach Aufwand die Wirtschaftlichkeit des Modells bewertet.

Grundsatz des systematischen Aufbaus Um ein Informationsmodell zu entwerfen, müssen seine Struktur und sein Verhalten beschrieben werden. Dabei bedarf es konsistenter Inter-Modellsichtbeziehungen, das heißt, bei der Modellierung einzelner Sichten müssen die Implikationen für andere Sichten betrachtet werden, damit eine sichtenübergreifende Integration ohne Konsistenzprobleme von statten gehen kann.

Grundsatz der Klarheit Es wird eine verständliche und einfache Konstruktion und Darstellung der Modelle gefordert, die allerdings adressatenindividuell ausfallen kann. Hilfsmittel dafür sind Hierarchisierung, Abstraktion, Festlegung von Darstellungsformen oder Filterungen,

welche anwenderspezifisch irrelevante Informationsobjekte ausblenden, um die Komplexität zu reduzieren.

Grundsatz der Vergleichbarkeit Um den Nutzen verschiedener Referenzmodelle diskutieren zu können, müssen diese verglichen werden können. Ein Vergleich ist zum einen auf Meta-Modellebene möglich, wobei die jeweiligen Beschreibungssprachen ineinander überführbar sein müssen, bevor Modellinhalte verglichen werden können. Beim Vergleich der Modelle selbst müssen die zugrunde liegenden Probleme und Anwendungsfälle in Relation gebracht werden können.

2.6.2 Bewertung der ITIL nach den GoM

Die in Abbildung 2-7 aufgeführten Ziele beim Einsatz von Referenzmodellen stehen letztlich auch hinter dem Interesse im IT Service Management, die ITIL zur Orientierung für Prozesstransformationen heran zu ziehen. Fraglich ist allerdings die Qualität der ITIL als adäquates Referenzmodell. Die Unsicherheit bei der Umsetzung von ITIL-Projekten und Missverständnisse bezüglich des durch die ITIL erzielbaren Nutzens deuten an, dass die ITIL für die unternehmerische Praxis kritisch hinterfragt werden muss [HZB04]. Hochstein untersucht als erster die ITIL, zum einen basierend auf Fallstudien in Großunternehmen, zum anderen nach den Grundsätzen ordnungsmäßiger Modellierung, und fasst seine Ergebnisse gemäß Tabelle 2-8 zusammen. Vereinfachend kann festgehalten werden, dass die ITIL für die Umsetzung von Service Management Prozessen zu ungenau bleibt.

2.6.3 Analoge Bewertung des MOF

Für das MOF ist bislang eine formale Bewertung unbekannt, sie wurde deshalb in dieser Arbeit durchgeführt. Die Aufarbeitung nach den GoM kann über die Ergebnisse von Hochstein [HZB04] und die Analogie des Microsoft Operations Frameworks zur ITIL für die sich überschneidenden Bereiche der beiden Frameworks gegebenenfalls verkürzt werden.

Genauso wie die ITIL verwendet das MOF keine künstliche Sprache, so dass die 6 Grundsätze nicht an Sprachregeln gezeigt, sondern nur allgemein bewertet und zum Teil an Beispielen gestützt werden können. Die Beispiele wurden derart ausgewählt, dass sie repräsentativen Charakter für das entsprechende Entscheidungskriterium aufweisen. Nach [Sch98] sind im Allgemeinen die Kriterien der Konstruktions- und Sprachadäquanz sowie der Grundsatz der Wirtschaftlichkeit für eine Bewertung besonders entscheidend. Wegen der Verwendung der natürlichen Sprache im MOF ist die Sprachadäquanz allerdings gering zu bewerten.

Grundsatz der Konstruktionsadäquanz im MOF Über die Übereinstimmung mit den ITIL Service Support und Delivery Prozessen trifft das *Vorliegen einer gemeinsamen Problemdefinition*, zunächst gegenüber der ITIL, zu. Für die SMFs im Operating Quadranten wird die Problemdefinition laut MOF auf die Microsoft Windows Server Administration erweitert. Allerdings sind auch diese Beschreibungen weitgehend herstellerunabhängig gehalten und können plattformübergreifend für die Server Administration zum Einsatz kommen. Aus diesem Grund und wegen der Anerkennung der im Grundsatz vergleichbaren ITIL in der IT-Branche, ist ein Konsens über die Problemdefinition gegeben.
Die fälschliche Verwendung des Begriffs der 'Best Practices' wird aus der ITIL übernommen und suggeriert gleichfalls die Bereitstellung von Verfahrensweisen, die vergleichbare Wettbewerbsvorteile verschaffen. Letzteres bleibt allerdings unbegründet beziehungsweise lassen sich durch die Vielfalt der MOF-Ausprägungen lediglich 'Common Practices' auf einer gemeinsamen MOF-Basis ausmachen.
Die *Inter-Modellkonsistenz* wird zwar in den MOF-Kapiteln zu den SMFs großteils über Beschreibungen der 'SMF-Relationships' vermittelt. Allerdings bleiben genaue Modellbeziehungen unklar. Speziell für das Configuration Management wurden in Appendix A die entsprechenden Beschreibungen aufgenommen. Darin lässt sich beispielhaft nachlesen, dass

Grundsätze	Kriterien	Erkenntnisse	Implikationen für das IT-Mgmt. /Handlungsempfehlungen
Konstruktions-adäquanz	Konsens über Problemdefinition	Kriterium ist erfüllt, aber falsche Verwendung des Begriffs 'Best Practices' (stattdessen enthält die ITIL 'Common Practices')	Missverständnis bzgl. der Erreichung komparativer Konkurrenzvorteile durch die ITIL
	Intra- und Inter-Modellkonsistenz	Wegen Verwendung von natürlicher Sprache nicht überprüfbar - Inkonsistenzen können deshalb nicht ausgeschlossen werden	Kritische Hinterfragung der Modellbeziehungen
	Berücksichtigung relevanter Informationsobjekte	Referenzszenarien fehlen	Erfahrungsaustausch (auf Konferenzen, Foren etc.) hilft bei der Filterung spezifisch relevanter Informationsobjekte)
	Minimalität	Kriterium ist nicht erfüllt	Nutzung von zusammenfassender Literatur unter selektiver Berücksichtigung der ITIL-Dokumentation
Sprachadäquanz	Semantische Mächtigkeit	Kriterium ist erfüllt	Keine Implikationen
	Formalisierungsgrad	Kriterium ist nicht erfüllt	Überprüfung der Sinnhaftigkeit unternehmensspezifischer Modellbeziehungen; Verzicht auf Interpretationen der ITIL; Nutzung formalsprachlicher, ITIL-basierender Referenzmodelle der Beratungsunternehmen; Forderung des Methodenpluralismus
	Sprachverständlichkeit	Kriterium ist erfüllt	Keine Implikationen
	Sprachrichtigkeit	Kriterium ist erfüllt	Keine Implikationen
Wirtschaftlichkeit	Kosten-Nutzen-Vergleich	Quantitative Berechnung kaum möglich, aber Erfahrungsberichte zeigen positiven ROI	Vielzahl an Nutzenkategorien wurde genannt; wichtigster Nutzenfaktor: Vermeidung des Risikos der Entwicklung eines ungeeigneten, nicht erfolgreichen ITSM-Modells.
	Robustheit	Kriterium ist erfüllt	Keine Implikationen
	Flexibilität	Kriterium ist erfüllt	Keine Implikationen
	Sprachadäquanz	Geringer Formalisierungsgrad	Zusätzlicher Aufwand bei der Systemspezifikation ist zu erwarten
	Übersetzbarkeit	Übersetzbarkeit in formalsprachliche Modelle ist kaum möglich	Zusätzlicher Aufwand bei der Erstellung des unternehmensspezifischen Prozessmodells zu erwarten
Systematischer Aufbau	Konsistente Inter-Modellsicht-beziehungen	Nur Verhaltensmodell vorhanden	Erfahrungsberichte anderer können helfen, Unsicherheiten, die aufgrund des fehlenden Referenzstrukturmodells entstehen, zu reduzieren
Klarheit	Hierarchisierung und Filterung	Kriterium ist nicht erfüllt	Nutzung filternder, strukturierender Literatur sowie auf ITIL basierender Referenzprozessmodelle
Vergleichbarkeit	Semantische Vergleichbarkeit	Eine semantische Vergleichbarkeit mit ITIL-basierten Referenzmodellen ist nicht gewährleistet	ITIL-basierte Referenzmodelle sind auf der einen Seite inhaltlich verkürzt und ergänzen auf der anderen Seite die ITIL mit spezifischen, meist auf Erfahrungen beruhenden Informationsobjekten.

Abbildung 2-8: Formale Bewertung des ITIL-Referenzmodells [HZB04]

es trotz der verallgemeinerten Darstellung unterschiedliche und damit inkonsistente Beschreibungen zu den Informationsobjekten innerhalb des Configuration Managements aus der Sicht einzelner SMFs gibt.
Wegen der vielfältigen Möglichkeiten der Auslegungen des MOF kann ebenfalls für die *Intra-Modellkonsistenz* keine allgemeine Aussage getroffen werden, wobei im Vergleich zur ITIL auffällt, dass MOF besser strukturiert ist und weniger Wiederholungen enthält. Diese Eigenschaften deuten auf eine vergleichsweise bessere Konsistenz hin.
Als Vorgriff der Auswertung möglicher CMDB-Inhalte in Kapitel 5 sei festgehalten, dass es keine *Berücksichtigung konkreter Informationsobjekte* gibt. Stattdessen werden vielfach, aber oberflächlich, Beispiele genannt, welche den Umfang möglicher Informationsobjekte vermuten lassen. Das ist für die Erfüllung des Kriteriums allerdings nicht ausreichend.
Die Erfüllung der Forderung nach *Minimalität* von Informationsobjekten kann mangels Festlegung letztendlich ebenfalls nicht zutreffen.

Grundsatz der Sprachadäquanz im MOF Durch die Verwendung der natürlichen Sprache ist die *semantische Mächtigkeit* ausreichend. Allerdings besteht gleichzeitig durch das Fehlen von Formalisierungen die Gefährdung durch Strukturvielfalt, Irregularität, Ambiguität und Vagheit (vgl. [HZB04]). Zwar bietet das MOF im Vergleich zur ITIL das übergreifende MOF-Prozessmodell mit untergeordneten, teils ausführlicheren Prozessdiagrammen. Diese sind zwar verständlich, entsprechen aber nicht der üblichen Modellierungspraxis, bleiben in der Anzahl zu gering und in der Aussage zu abstrakt, als dass ein gesteigerter *Formalisierungsgrad* abgeleitet werden könnte.
Die *Sprachverständlichkeit* und *Sprachrichtigkeit* ist über die Verwendung der natürlichen Sprache gegeben.

Grundsatz der Wirtschaftlichkeit im MOF Ein *Kosten-Nutzen Vergleich* des MOF ist insofern schwierig vorzunehmen, als dass grundsätzlich durch das hohe Abstraktionsniveau individuelle Modellausprägungen zu erarbeiten sind und Kosten/Nutzen von den Eigenschaften der Ausprägungen abhängen, also nicht mittelbar auf das MOF bezogen werden können. Die Tatsache der Notwendigkeit, Ausprägungen abzuleiten, deutet allerdings bereits auf einen geringeren, unmittelbaren Nutzen und höhere Einführungskosten hin. Die Gleichartigkeit zur ITIL und deren branchenübliche Verwendung sprechen aber für ein positives Kosten-Nutzen-Verhältnis.
Das MOF ist inzwischen in Version 3 erschienen, unterlag also schon gewissen Veränderungen. Dem Autor liegen Version 1 und 2 für eine Untersuchung nicht vor, deshalb wird vereinfachend analog zur ITIL davon ausgegangen, dass durch die Fortentwicklung die Validität grundsätzlicher Modellregeln nicht außer Kraft gesetzt wurde. Dadurch ist das Kritierium der *Robustheit* des Modells erfüllt.
Die *Flexibilität* des Modells ist zum einen durch die natürliche Sprache, zum anderen durch die hohe Abstraktionsebene gegeben.
Die *Sprachadäquanz* unter wirtschaftlichen Gesichtspunkten ist zum einen durch die der Verständlichkeit zuträglichen Verwendung der natürlichen Sprache gegeben. Zum anderen ist jedoch die individuelle Modellableitung, wie oben bereits genannt, mit zusätzlichem Aufwand verbunden. Andere in natürlicher Sprache beschriebenen Modelle können wesentlich genauere Vorgaben als das MOF liefern - insofern ist Sprachadäquanz unter wirtschaftlichen Gesichtspunkten mangels Formalisierung vergleichsweise schlecht.
Die *Übersetzbarkeit* in eine andere Modellsprache ist wegen der geringen Formalisierung nicht möglich.

Grundsatz des systematischen Aufbaus im MOF Das erweiterte MOF-Prozessmodell setzt die SMFs über die Anwendbarkeit innerhalb eines Optimierungszyklus in Verbindung und ordnet verschiedene Teamrollen zu, die im Team-Modell zusammengefasst sind. Die Forderung von *konsistenten inter-Modellsichtbeziehungen* wird über die beiden übergreifenden MOF-Modelle - auf oberster Ebene - nachgekommen. Auch sind die Kapitel der MOF-Beschreibungen weitgehend ähnlich aufgebaut (Process and Activities, Roles and Responsi-

bilities, Relationships to other Processes), so dass im Vergleich zur ITIL mehr Systematik enthalten ist.

Grundsatz der Klarheit im MOF Zwar gibt MOF viele Ideen und Appelle zur Umsetzung, deren Ausgestaltung bleibt allerdings unklar. Hierzu muss bemerkt werden, dass aus der Arbeitspraxis der in der MOF genannte Bedarf an Funktionen dem Leser im gewöhnlichen bekannt ist. Wünschenswert wären allerdings Hinweise, wie genau diesem Bedarf mit 'Best Practices' für eine Umsetzung nachgekommen werden kann, strukturiert nach verschiedenen Anforderungen. Für die SMFs im Operating Quadranten kommt das MOF hier zwar streckenweise nach, bleibt aber insgesamt zu ungenau, weshalb der Grundsatz der *Klarheit* nicht erfüllt ist. Im Vergleich mit der ITIL fällt allerdings auf, dass das MOF Vorschläge besser zusammenfasst. Letztendlich sei an dieser Stelle nochmals darauf hingewiesen, dass das MOF dazu beiträgt, Klarheit über die Begriffsverwendung und die Aufteilung der SMFs zu schaffen, was die Kommunikation unter Fachabteilungen deutlich erleichtert.

Grundsatz der Vergleichbarkeit im MOF Als Folge der Ableitung des MOF aus der ITIL, die sich über die grossen Ähnlichkeiten der obigen Ergebnisse mit der ITIL-Bewertung nach den GoM von Hochstein (2-8) bestätigt, ist *Vergleichbarkeit* mit der ITIL für die sich überschneidenden Bestandteile gegeben. Mangels Formalisierung sind zwar keine harten Vergleiche möglich, dennoch lassen sich grundsätzliche Ziele und Aktivitäten gegenüberstellen.

Fazit: Im Vergleich zur Bewertung der ITIL (2-8) ergeben sich für das MOF nur geringfügige Unterschiede in der Qualität der Modellierung nach den GoM. Festzuhalten ist, dass das MOF besser strukturiert ist, Inhalte weniger weitschweifig darstellt und für das Operating mehr individuelle Beschreibungen liefert. Wegen der hohen Ähnlichkeit werden die beiden Frameworks in den folgenden Ausführungen als einheitliche Modellbasis betrachtet, wobei die ITIL meist vorrangig genannt wird.

Fasst man die formale Bewertung der ITIL in Tabelle 2-8 zusammen, ergibt sich auf der Seite der Stärken die Bereitstellung eines ITSM Prozessmodells, das aufgrund von Erfahrungen bei Verwendung gewöhnlich zu einen positiven ROI verhilft. Auf der Seite der Schwächen ist anzuführen, dass unter anderem aufgrund von nicht auszuschließenden Inkonsistenzen, fehlender Formalisierung und fehlenden Referenzszenarien ein *Mangel an Klarheit* besteht. Vor dem Ziel der Strukturierung und Konkretisierung innerhalb dieser Arbeit begründet sich damit die Notwendigkeit modellkonformer Festlegungen, um detailliertere Vorschläge für eine Umsetzung des Configuration Managements ausarbeiten zu können.

2.7 Input-/Output-Daten für ITIL-Prozesse

Nachdem nun ITIL und MOF im Überblick mit einer Bewertung der Referenzmodellqualität vorgestellt wurden, soll nun im folgenden Kapitel, zur Überleitung auf die detaillierte Beschreibung des Configuration Mangements, die Bedeutung der Konfigurationsdaten der Infrastrukturkomponenten für ITSM Prozesse im Vergleich zu anderen Servicemanagementdaten heraus gearbeitet werden. Zur Vereinfachung geschieht dies mit Beschränkung auf die ITIL Service Support und Delivery Prozesse.

Gemäß der Bewertung nach den GoM, lässt die hohe Abstraktionsebene der ITSM Prozessbeschreibungen lediglich die Nennung von Datenklassen zu, die in den Prozessen benötigt oder ausgegeben werden. Eine Detaillierung der Daten ist - in Grenzen - lediglich bei unternehmensspezifischen Ausprägungen der ITSM Prozesse möglich. Die Input- (I) und Output-Datenklassen (O), die in Tabelle 2-9 den ITIL Service Support und Delivery Prozessen zugeordnet werden, wurden nach einer Auswertung von Aufstellungen in [Vog04] und [Vog02] zusammengeführt. Dabei wurden zum großen Teil die verwendeten Begriffe der Datenklassen beibehalten - im Glossar unter B sind die unbeschriebenen Akronyme nachzulesen. Die Tabelle gibt einen Überblick über den Umfang des Informationsbedarfs und der Informationsbereitstellung in den ITIL Prozessen. Mit Hinblick auf

das später aufgenommene Szenario lässt sich aufgrund der vielfältigen Verknüpfungen der Daten ableiten, dass zur Reduzierung des Kommunikationsbedarfs zum Austausch gewisser Informationen eine Rollen- beziehungsweise Prozess-übergreifende Aufgabenzuweisung auf IT-Mitarbeiter durchaus sinnvoll ist. Des Weiteren ist es für eine Datenhaltung der verknüpften Daten des Servicemanagements - gemäß einem integrierten Managementansatz - offensichtlich erstrebenswert, die Referenzen innerhalb der Datenklassen zu nutzen, um Vorteile wie Abfragen auf den gesamten Datenbestand oder die Vermeidung von doppelter Datenpflege zu nutzen.
Diejenigen Datenklassen, welche die zentrale Funktionalität des Configuration Managements betreffen, sind in der Tabelle grau hinterlegt. Das Configuration Management hält in der CMDB *Informationen über Konfigurationselemente* vor, die in allen anderen ITSM Prozessen benötigt werden. Das Change-, Release- und Service-Level-Management tragen darüber hinaus, mit der Bereitstellung von Konfigurationsdaten, zur Einbringung von Änderungen in die CMDB bei. *Informationsbedarf an Konfigurationselementen* wird von fast allen ITSM Prozessen an das Configuration Management gemeldet. Befinden sich die angeforderten Daten nicht in der CMDB, kann das Configuration Management für die Bereitstellung der fehlenden Daten sorgen. Insofern ist diese Datenklasse für die Optimierung des Configuration Managements ein wichtiger Bestandteil.
Die *Impact*-Analyse ist zur Erkennung von Risiken und damit für proaktives Management von zentraler Bedeutung. Auswirkungen lassen sich beispielsweise über Daten des Capacity Managements abschätzen, etwa die voraussichtlichen Folgen einer zu geringen Kapazitätsausweitung auf die Produktionsmöglichkeiten, oder das Service-Level-Management, welches beispielsweise die Vertragsstrafe für die Nichterbringung vereinbarter Leistungen beschreibt. Die CMDB stellt mit Abhängigkeitsinformationen innerhalb der IT-Infrastruktur gleichfalls wichtige Daten zur Impact-Analyse bereit. So können beispielsweise die Auswirkungen von Changes auf die IT-Infrastruktur besser abgeschätzt oder die kritischen Dienstbestandteile abgefragt werden, welche mit einem gewissen Risiko die Verletzung eines Service Level Agreements verursachen können. Zur Ergänzung dieser Auswertung bezüglich der Art der angeforderten Informationen über Konfigurationselemente wird in Kapitel 5.2.1 eine detailliertere Untersuchung der Anfragen aus MOF Service Management Functions an die CMDB beschrieben.

	Incident Management	Problem Management	Change Management	Release Management	Configuration Management	Service Level Management	Capacity Management	Availability Management	Continuity Management	Financial Management
Capacity Plans				I	I		O	I	I	I
Capacity/Availability Events/Alerts	O	O					I	I		
Change Records			O							
Change-Plan (FSC)	I		O		I		I	I	I	
Continuity Plans			I		I				O	
Finanzinformationen			O			IO				IO
Geschäftsentwicklung und -ziele	I	I	I	I	I	I	I	I	I	I
Impacts	I	I	IO	I	O	IO	O	I	I	
Incident Records	O						I	I		
Informationen über Konfigurationselemente	I	I	IO	IO	IO	IO	I	I	I	I
Informationsbedarf bzgl. der CMDB	O	O	O	O	I	O	O	O	O	
Kosten der Leistungserstellung	I	I	IO	I	I	IO	I	I	I	IO
Kundenanforderungen/-zufriedenheit						I				
Notfallpläne	I	I				I			O	
Performance/Capacity Informationen				I	I		IO			
Post Implementation Review (PIR)		I	O							
Probleme/Bekannte Fehler	I	IO								
Rechnungen										O
Release Policy/Plans/Notes	I	I	I	O			I	I	I	
Reports	O	O	IO		O	I	O		O	
Requests for Change (RFCs)	O	O	IO	I	IO		O			
Rollout-Informationen	I			IO		I	I			
Schulungsinformationen				O						
Schwachstellen und Risikoprofile		I					O	IO	I	I
Schwellenwerte/Checkpoints	I						O	O		
Service Improvement Programme						O				
Servicekatalog	I				I	IO				
Störungsinformationen	IO	I			O					
Supportvereinbarungen (SLAs, OLAs, UCs)	I	I				IO				I
Testpläne und -ergebnisse			I	IO					O	
(Umgehungs-)Lösungen	IO	IO								
Verfügbarkeits- und Leistungsanforderungen						O	I	I	I	
Verfügbarkeitsplan								O	O	
Vorbeugemaßnahmen		O								
Wartungspläne			I					O		

Abbildung 2-9: Input-/Outputdaten der zehn zentralen ITIL-Prozesse

2.8 Die zentrale Rolle des Configuration Managements

Mangels Überschneidungsfreiheit, Vollständigkeit und Gleichwertigkeit der in Tabelle 2-9 aufgelisteten Datenklassen, lassen sich zwar nur tendenzielle Schlüsse ziehen, diese unterstützen allerdings die empirischen Ergebnisse aus der Beratungspraxis [EXP]. Die hohe Anzahl an ITSM Prozessen mit Inputanforderungen, unter der Klasse 'Informationen über Konfigurationselemente', zeigt die fundamentale Rolle des Configuration Managements, beziehungsweise des Betriebs einer CMDB für andere ITSM Prozesse. Die Bedeutung des Configuration Managements wird von IT-Verantwortlichen auch dementsprechend hoch bewertet. In einer Befragung deutscher Unternehmen, denen der ITIL-Standard bekannt ist, hat die ComConsult GmbH Anfang 2004 die Frage gestellt: 'Welche operationellen Prozesse nach ITIL halten Sie für besonders wichtig (Mehrfachnennungen möglich)?', vgl. Abbildung 2-10.

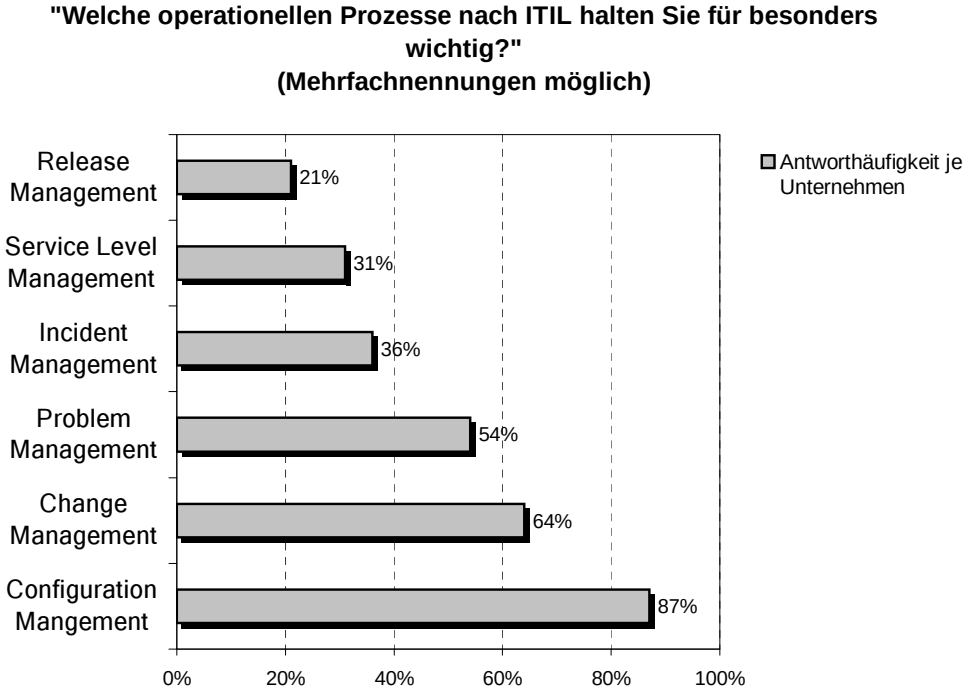


Abbildung 2-10: Bedeutung des Configuration Managements [Nic04]

Bedingt durch den größten, unmittelbaren Bedarf und die einfache Umsetzung, werden in vielen IT-Abteilungen anfangs spezifische Toollösungen für ITSM Prozesse wie das Incident- und das Problem-Management eingeführt. Configuration Management wird zwar häufig bereits in kleineren IT-Umgebungen betrieben - allerdings nicht mit entsprechenden CMDB-Tools. Stattdessen dokumentieren üblicherweise Systemadministratoren, mit dem gesammelten Wissen über den eigenen Zuständigkeitsbereich, insbesondere systemorientierte Konfigurationsdaten in Einzeldokumenten [Vog02]. Häufig ist diese Art der Dokumentation in einer überschaubaren IT-Abteilung mit einem beständigen Mitarbeiterstamm ausreichend. Der Koordinationsbedarf wächst allerdings, wenn eine IT-Abteilung durch ihre Größe für den einzelnen IT-Mitarbeiter unüberschaubar wird, wenn örtlich verteilt, zeitlich versetzt oder mit häufig wechselnden Mitarbeitern gearbeitet wird, oder wenige Standardprodukte, dafür viele, individuelle Entwicklungen, zum Einsatz kommen. Fehlt in derartigen Umgebungen eine CMDB, welche die Verknüpfung von Konfigurationselementen mit anderen Daten des Servicemanagements zulässt, so muss im Bedarfsfall eine individuelle Recherche an der IT-Infrastruktur selbst vorgenommen werden, die gerade bei vielen Zuständigkeiten und Zugriffsbeschränkungen sehr aufwändig werden kann.
Folgende Beispiele sollen den Einsatz von Konfigurationsdaten aus einer CMDB in anfragenden ITSM Prozessen verdeutlichen. Das Change-Management, das die bereits genannte Abhängigkeits-

analyse zur Risikobewertung eines Changes nutzt, kann über eine CMDB diejenigen Benutzer und Administratoren ausfindig machen, die von einem Change betroffen sind. Damit können Informationen, wie beispielsweise der Change-Zeitpunkt, die wahrscheinliche Ausfalldauer eines Dienstes und gegebenenfalls vorzunehmende Änderungen nach dem Change zielgerichtet an die Betroffenen verteilt werden. Das Financial Management benötigt Informationen über die Wirtschaftsgüter in der IT und erhält diese aus der mit dem Asset-Management kombinierten CMDB. Im Capacity-, Continuity-, Availability-, und SLA-Management müssen Serviceziele mitunter an Konfigurationselementen festgemacht werden. So können beispielsweise zur Überwachung von Kapazitäten, auf ständig aktualisierte Komponentenstati in der CMDB, Schwellwerte gesetzt werden, welche bei Überschreitung eine Warnaktion auslösen. Die Zuordnung von Incidents und Problems zu Konfigurationselementen ermöglicht es, doppelte Erfassungen zu erkennen, Probleme zu korrelieren und eine allgemeine Lösung bereit zu stellen. Die Lizenzverwaltung für Software kann über eine Zuordnung von Lizenzen zu Konfigurationselementen, wie Rechensysteme oder Benutzer, automatisch aktualisiert werden, wenn die tatsächliche Nutzung eingestellt wird. Es gibt über die genannten Beispiele hinaus eine Reihe weiterer, zweckdienlicher Verknüpfungen, die ausschlaggebend sind für das Interesse an Configuration Management. Letzteres wird nun im folgenden Kapitel nach ITIL und MOF beschrieben.

3 Configuration Management nach ITIL und MOF

Im folgenden Kapitel werden nach einem Überblick über die Aufgaben und den Nutzen des Configuration Managements, die Anforderungen aus ITIL und MOF detailliert vermittelt. Es folgt ein Ansatz der unmittelbaren Ableitung eines Informationsmodells aus den Anforderungen und die Darstellung kritischer Faktoren bei Einführung und Betrieb des Configuration Managements.

3.1 Aufgaben und Nutzen des Configuration Managements

Die Aufgaben des Configuration Managements im laufenden Betrieb umfassen (vgl. Abbildung 3-1):

- **Identifikation von Configuration Items** (CIs) und Erfassung in der Configuration Management Database (CMDB) mit Verantwortlichkeiten, Status-Informationen, Abhängigkeiten mit anderen CIs und sonstigen Konfigurationsdaten.
- **Kontrolle von Änderungen**, damit nur autorisierte Changes in die CMDB eingebracht werden.
- **Durchführung von Audits** zum Abgleich der IST-Daten der IT-Infrastruktur mit dem SOLL-Modell in der CMDB. Diese Aufgabe beinhaltet das ständige Einbringen von Änderungen der CI-Stati.
- **Generierung von Reports** zum Configuration Management.
- **Prüfung des Datenmodells der CMDB** und gegebenenfalls dessen Anpassung.
- **Bereitstellung von Konfigurationsdaten** für andere ITSM Prozesse.

Folgende Vorteile ergeben sich durch die Einführung des Configuration Managements [Som04]:

- Kontrolle über die IT-Infrastruktur, denn nur bekannte Komponenten können kontrolliert und eingesetzt werden.
- Ein erfassbares Modell der Infrastruktur ist nicht nur IT-intern sondern auch gegenüber Außenstehenden, etwa Zulieferern, vorteilhaft für eine bessere Kommunikation.
- Verbessertes Asset-Management durch die Zuordnung von Abhängigkeiten unter den CIs und aktuelle Informationen über deren Zustand oder Verfügbarkeit, beispielsweise zur Identifikation defekter oder entwendeter Assets.
- Reduziertes Risiko durch Changes und deren beschleunigte Durchführung, nachdem Abhängigkeiten unter den CIs die Auswirkungen der Changes abschätzbar machen.
- Verbesserte Kostenplanung über die Prüfung der Nutzung einzelner CIs und den darauf verwendeten Administrationsaufwand.
- Unterstützung der zugreifenden ITSM Prozesse mit Konfigurationsdaten, vgl. Kapitel 2.7.

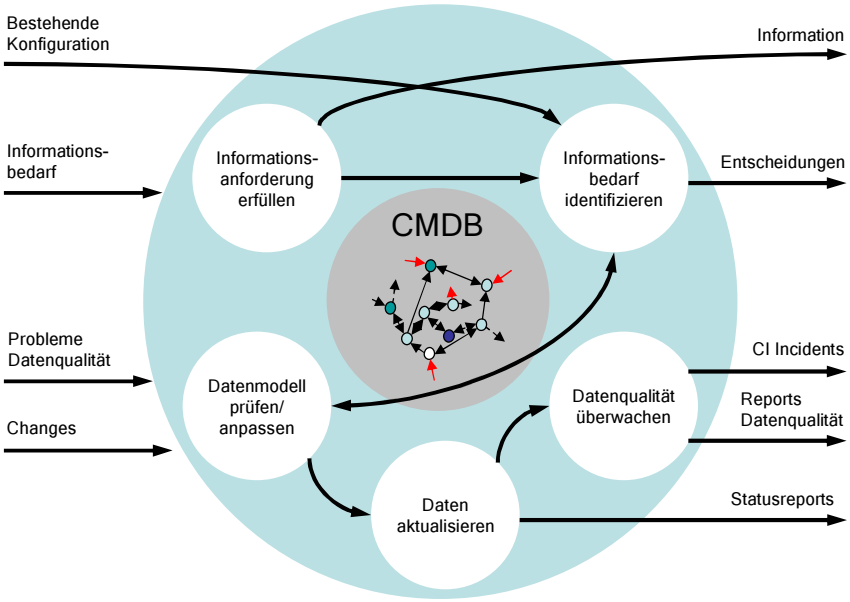


Abbildung 3-1: Aufgaben des Configuration Managements nach [Vog02]

3.2 Anforderungsanalyse

Statt die ITIL und das MOF mit ihren vielfältigen Empfehlungen an dieser Stelle neu aufzubereiten, werden in diesem Kapitel, mit dem Ziel einer Umsetzung, nur die wichtigsten Anforderungen der beiden Frameworks dargelegt. Es versteht sich von selbst, dass eine Zusammenfassung immer Informationsverlust mit sich bringt. Es wurde allerdings darauf geachtet, die Essenz der Aussagen der beiden Frameworks zum Configuration Management nach den folgenden Kriterien zu extrahieren.

3.2.1 Methodik

Folgende Auswahlkriterien wurden in den offenen Beschreibungen der ITIL und des MOF gewählt, wobei anzumerken ist, dass die eingesetzte, natürliche Sprache bedauerlicherweise keine härteren Kriterien zulässt.

- Zentrale Nennung: Anforderungen, die in Überschriften oder in zentralen Kapiteln aufgenommen wurden, wird mehr Bedeutung beigemessen.
- Wiederholungen: die ITIL strukturiert Empfehlungen zum Configuration Management zwar nach Kapiteln wie etwa zu Prozessaktivitäten, Toolauswahl und Assoziationen mit anderen Prozessen, stellt aber innerhalb der Kapitel keine klare Abgrenzung zu anderen Themen her, sondern schafft viele Querbezüge. Dadurch ergeben sich häufig Wiederholungen, wodurch sich eine gewisse Gewichtigkeit des entsprechenden Aspekts vermuten lässt.
- Anforderungen vs. Empfehlungen: Anforderungen (beispielsweise 's.th. should be done') können von Empfehlungen (beispielsweise 's.th. could be done') unterschieden werden. Allerdings ist auch dieses kein klares Kriterium, weil mangels Deutlichkeit der Formulierungen häufig nicht dazwischen unterschieden werden kann beziehungsweise unklar bleibt, aus welchen Gesichtsgründen die ITIL zum einen Anforderungen, zum anderen Empfehlung formuliert.

Diese Unterscheidungsversuche zeigen in Ergänzung der Bewertung nach den GoB, dass der ITIL wegen ungenauer Beschreibungen und der losen Sammlung von Beispielen aus Common-Practices, die Eigenschaft der Ableitbarkeit einer umsetzbaren Spezifikation fehlt. Diese Tatsache wird später

auch an einem Ansatz der Ableitung eines Informationsmodells für eine CMDB aus den ITIL-Vorschlägen bestätigt. Das besser strukturierte MOF-Kapitel zum 'Configuration Management' wurde danach ausgewertet, welche ergänzenden Anforderungen bestehen. Es reduziert die ITIL-Vorschläge auf Kernaussagen und bestätigt durch Übereinstimmung die getroffene Auswahl der ITIL-Anforderungen. Allerdings werden die ungenauen Beschreibungen der ITIL auch hier nicht näher konkretisiert, was es später erforderlich machen wird, zweckdienliche, eigene Festlegungen zum Configuration Management zu treffen.

3.2.2 Anforderungskatalog

Um dem Leser ein bestmögliches Verständnis zu verschaffen, werden im Folgenden die Anforderungen aus dem ITIL-Buch 'Service Support' der OGC [OGC00], Kapitel 'Configuration Management' nach den oben genannten Kriterien zusammengefasst und um die zusätzlichen Anforderungen des MOF [Mic04c] ergänzt. Letztere sind zur Unterscheidung kursiv formatiert. Die Seitenangaben beziehen sich auf die beiden genannten Quellen. Die Zusammenfassung folgt ausdrücklich keinem Standard der Requirement Analysis, nachdem einige üblicherweise geforderten Bestandteile für ein 'Requirements Analysis Document' in den beiden Frameworks nicht beschrieben werden. Stattdessen beschränkt sich der Autor auf eine Darstellung der Ziele des Configuration Managements, der Systemumgebung für den Einsatz von Configuration Management und die nach Prozessaktivitäten gegliederte Auflistung der Anforderungen.

Zieldefinition für das Configuration Management
Einführung und Verwaltung eines logischen Modells der IT-Infrastruktur als Informationsmodell für Konfigurationsanfragen vor dem Optimierungsproblem 'maximum control with minimum records', S. 156. *Sicherstellung, dass nur autorisierte Komponenten verwendet werden, S.1*

Systemumgebung
Verwendung dort, wo die dezentrale und unkoordinierte Verwaltung von Konfigurationen einer heterogenen IT-Infrastruktur für ein IT-Management durch mehrere Personen effizienzmindernd ist.

Begriffsklärung

- Configuration Item (CI): eine IT-Komponente, die in Aktivitäten des Configuration Managements einbezogen ist. Jedes CI kann aus anderen CIs bestehen. Beispiele für CIs sind die Komponenten eines Servicegraphen 2-2 [Mic04g]. CIs können stark in der Komplexität, Größe und im Typ abweichen [Mic04c].

- Configuration Management Database (CMDB): Eine Datenbank, die alle relevanten Details jedes CIs wie auch Abhängigkeiten der CIs untereinander enthält [Mic04c].

- Configuration Manager: Die Rolle des Verantwortlichen für das Configuration Management. Diese Rolle stellt gegebenenfalls die Gruppe der untergeordneten Configuration Manager, genannt 'Configuration Management Staff', zusammen und koordiniert deren Aktivitäten [Mic04c].

- Key Performance Indicator (KPI): Schlüsselkennzahl zur Messung der Zielerreichung [Mic04c].

- Definitive Software Library (DSL): gesicherter Aufbewahrungsort, in welchem die freigegebenen Versionen aller Software Komponenten geschützt gespeichert werden [Vog02].

- Definitive Hardware Store (DHS): Physikalische Sammlung aller im IT-Betrieb verwendeten Hardware(-Komponenten) im Sinne eines Lagers. Der DHS ist parallel zum DSL und zur ODL zu sehen [DKW04].

- Operational Documentation Library (ODL): Zentrale, logistische Informationssammlung im Sinne einer Bücherei des IT-Betriebes. Parallel zum DHS und der DSL beinhaltet die ODL alle physikalisch existierenden Dokumentationen. Die ODL liefert die genauen Informationen zu allen Betriebsdokumenten, -prozessen und -verfahren, speziell Übergabeverfahren. Des Weiteren sind hier alle technischen Dokumentationen wie Manuals, Specification Sheets, Systemkonzepte, Netz-, Raum- und Kabelpläne niedergelegt [DKW04].

Anforderungen

1. Planung und Strategie
   a) Zieldefinition und Prozessdefinition, S. 129
      i. Planung eines Optimierungsprozesses mit ständiger Überwachung der Prozessbestandteile, der Arbeitslast und des Wachstums des Configuration Managements, S. 149; Prüfung geänderter Anforderungen an die CMDB: S. 140, 149, 150.
      ii. Ziel ist die ideale Unterstützung der zugreifenden IT Service Management Prozesse, S. 121 mit starker Zweckorientierung, S. 135 (Beschleunigung der Arbeitsprozesse, Schaffung von Kapazitäten für Erweiterungen).
      iii. Schrittweises Vorgehen der Umsetzung, S. 128, mit Orientierung am größten Verbesserungsbedarf, S. 135, beginnend mit 'well-defined service and corporate data', S. 128 und Analyse der Informationsanfragen an die IT-Infrastruktur.
      iv. Vereinfachung der Datenhaltung durch Komplexitätsreduzierung, bsp. durch Standardisierung, Automatisierung, Reduzierung des Umfangs, Verteilung der Datenpflege auf mehrere Verantwortliche, Reduzierung lästiger Routinearbeit zur Überprüfung des Datenmodells, S. 127.
   b) Bedarfserhebung
      i. Die CMDB wird häufig für SLA-Managements (mit Limits auf den Kenngrößen der Konfigurationen) sowie für die Anlagenbuchhaltung der Anlagegüter in der IT-Infrastruktur genutzt, S. 124.
      ii. Eine starke Bindung zu Finanzplanung, Administration und Einkauf sollte gegeben sein, S.153.
   c) Aufbau eines Datenmodells der CMDB
      i. Inhalte: Configuration Items (CIs) sind Hardware, Software und Dokumentationen, S. 122 mit ihren physischen und funktionalen Spezifikationen, S. 129, Prozessbeschreibungen, S.1. Weitere Beispiele dazu sind auf S. 137 aufgeführt, darunter Benutzer, Zulieferer und Verträge .
      ii. Abhängigkeiten
         A. zwischen CIs, S. 122, Incident Records, Problems, Errors, RFCs, S. 141, Unternehmensinformationen über Mitarbeiter, Lieferanten (mit Berücksichtigung des Datenschutzes), Ortsinformationen und Abteilungen, S. 124, für den Zweck der Abhängigkeitsanalyse, S. 153 .
         B. Hierarchischer Aufbau der CIs, S. 122, 141 .
         C. Aufnahme weiterer Abhängigkeiten, beispielsweise 'connected to', 'uses another', S. 141.
         D. 'Configuration structures' zur Beschreibung der Beziehung und Position eines CIs innerhalb der Strukturen sollten entwickelt werden, S. 138.
      iii. Definition von Attributen, Beispiele auf S. 164
         A. CI-Owner regelt Verantwortlichkeit, S. 122 .
         B. Life-Cycle-Stati sollten typ-spezifisch definiert werden, sofern sinnvoll (Beispiel für ein Release: registered, accepted, installed, withdrawn), S. 141 .

iv. CI-Typen und -Gruppen ermöglichen die Reduzierung der Verwaltung von Abhängigkeiten einzelner CIs, sofern diese auf Typen/Gruppen bezogen werden können, S.140, 141.

v. Aufnahme von CIs unterschiedlicher Komplexität, S. 153; Definition der Detailtiefe pro CI-Typ je nach Bedarf: Analyse über Informationsanfragen an die CMDB, mit dem Zweck, einerseits ein verwaltbares, da reduziertes Datenmodell zu schaffen, zum anderen eine aussagekräftige Informationsbasis aufzubauen, S. 126, 140.

vi. Möglich ist die Priorisierung von CIs, um die Dringlichkeit der Überwachung zu berücksichtigen (bsp. Server mit höherer Priorität als Desktops), S. 129 .

vii. Aufnahme einer CI-Historie, S. 129, S. 147.

viii. Aufnahme einer Zugriffskontrolle, S. 144 auf einer 'need-to-know-Basis', S. 153.

ix. Einplanung von Configuration Baselines als Wiederherstellungspunkte, S. 123, 142.

d) Ergänzende Libraries

i. Forderung der Einrichtung einer Softwarebibliothek (DSL) in Ergänzung zur IT-Infrastrukturverwaltung der CMDB, S. 154.

ii. Die DSL dient zur Aufbewahrung von Masterkopien (physisch und logisch), S. 125, für Versionsverwaltung und Lizenzmanagement, S. 146.

iii. Die DSL wird durch Prozesse aus dem Software Configuration Management verwaltet, S. 144.

iv. Einrichtung eines Definitive Hardware Store (DHS).

v. Verwaltung einer Dokumentenlibrary (ODL: Operational Documentation Library).

vi. Für diese Libraries gilt: Verwaltung von Bezügen zu CIs und Zugriffskontrolle, S. 142, sowie gleiche Audits wie für die CMDB.

e) Toolauswahl

i. Forderung einer reichen Abfragesprache auf die CMDB, S. 124, 154.

ii. Forderung einer hohen Flexibilität bezüglich des Datenaustausches mit anderen CMDBs ohne 'Rekeying', S. 128.

iii. Integration der Verwaltung des Entwicklungssystems parallel zum Produktivsystems, S. 128.

iv. Zusammenstellung von Tools zur automatischen Datenerfassung und Implementierung von CMDB-Schnittstellen, S. 124. Idealerweise sollte die CMDB automatisch aktualisiert werden, sobald sich der Status von CIs ändert, S. 145, 153.

f) Personalmanagement

i. Rollenzuweisung und Training, S. 160, Redundanzen innerhalb der Rollen für höhere Verfügbarkeit des Configuration Managements, S. 161.

ii. Configuration Management darf nur durch trainierte und autorisierte Configuration Manager sowie untergeordnetem Configuration Management Staff durchgeführt werden, S. 129, die in den Entwicklungsprozess einbezogen werden sollen und für die Gültigkeit der Daten (in festzulegenden Teilbereichen) der CMDB Verantwortung übernehmen, S. 131. *MOF nennt auf S.10 Rollen des MOF-Team Modells, die mit einbezogen werden sollen.*

iii. Bereitstellung des Zugriffs auf die CMDB soweit als möglich, S. 129, beispielsweise auch für externe IT-Partner mit Vermittlung des Configuration Management Plans, S. 159.

g) Definition von KPIs, um die Effizienz des Configuration Managements bei regelmäßigen 'Audits' zu messen, S. 150.

h) Dokumentation des Configuration Managements im Configuration Management Plan.

2. Identifikation von CIs
   a) Identifizierung mit Labeling und Versionsstandserfassung, S. 129, 143.
   b) Verwendung von Namenskonventionen, aus denen sich hierarchische Bezüge zu anderen CIs sowie eine zeitliche Ordnung via Versionsnummer ableiten lassen, S. 143.
   c) Einführung von CI-Labels für physische Items, die schnell erfassbar sind (für Reviews und Wartung), S. 143.
   d) Redundante CIs sollen systematisch in der CMDB gelöscht werden, S. 149.
3. Kontrollierte Änderungen der Informationen in der CMDB
   a) Betrachtung von Change-, Release- und Configuration Management als Einheit, S. 152. Das Change-Management autorisiert über die Bereitstellung von 'Requests for Changes' einen CMDB-Change, S. 145, mit Bezügen zu Verantwortlichen, S. 147.
   b) Registrierung von CIs in der CMDB mit der (Produkt-)Bestellung oder internen Bereitstellung, S. 144.
   c) Automatisierung trägt signifikant zur Effizienz- und Effektivitätssteigerung bei und sollte wo möglich integriert werden (Fehler- und Kostenminimierung), S. 124, 154.
      i. Automatische Abhängigkeitserfassung, sofern möglich, bei Aufnahme neuer CIs, S. 153.
      ii. Automatisches Updaten der Versionsnummer eines CIs bei Änderung, S. 154 .
   d) Absicherung der CMDB: Zugriffs- und Betriebssicherheit, S. 146, 149.
4. Statusüberwachung, Verifizierung und Audits
   a) Die regelmäßige oder angeforderte Überwachung ist mitunter Aufgabe des Configuration Management Staffs, S. 147, 148, *mit Aufzeichnung des Datums der letzten Verifizierung, S. 26.*
   b) Damit wird der Forderung nach Bereitstellung von korrekten und exakten ('accurate') Daten, S. 129, nachgekommen.
   c) *Falls es keine Übereinstimmung der CMDB-Einträge mit dem Produktivsystem gibt, soll ein Incident zur Korrektur generiert werden, S. 27.*
5. Reporting: Regelmäßige Generierung von KPI-Reports über den Erfolg des Configuration Managements beziehungsweise Problemstellungen im Betrieb (Beispiel: festgestellte Abweichungen zwischen IST- und SOLL-Modell, die unautorisierten Änderungen zuzurechnen sind), S. 150.

Auffällig an der Auflistung der Anforderungen sind die wenigen Ergänzungen, die im entsprechenden MOF-Kapitel besonders hervorgehoben sind und deshalb hinzugenommen wurden. Hintergrund dafür ist, dass sich das MOF insbesondere für das Operating zur Aufgabe gemacht hat, Prozessbeschreibungen der ITIL hinzuzufügen, innerhalb der Bereiche des Service Support und der Service Delivery jedoch die ITIL übernimmt. Zwar orientieren sich beide Frameworks bei der Darstellung des Configuration Managements an den Management-Aktivitäten, das ITIL-Kapitel ist allerdings weniger geordnet als die etwas gekürzte Darstellung innerhalb des MOFs.

3.3 Ansatz zur Ableitung von Managementmodellen

Nachdem ITIL und MOF nach den Grundsätzen ordnungsmäßiger Modellierung bedauerlicherweise Mängel in der Qualität der Referenzmodellierung aufweisen, ist eine formale Ableitung eines Konzepts nicht möglich. Stattdessen müssen im weiteren Verlauf verschiedene Möglichkeiten der Strukturierung genutzt werden, um über *schrittweise Eingrenzungen* die Konkretisierung eines

umsetzbaren Konzepts plausibel zu machen.

Um die Anforderungen nun weitergehend zu strukturieren, kann auf die Modelle des OSI-Managements zurück gegriffen werden. Das OSI-Management bezieht sich auf technische Managementebenen. Allerdings spricht nichts dagegen, die unter 2.2 vorgestellten Grundprinzipien bezüglich der Submodelle des OSI-Managements auch auf der Ebene des Service Managements anzuwenden, wobei technische Aspekte des OSI-Managements hier natürlich entfallen. Diese Anwendung soll zeigen, inwieweit ITIL und MOF für das Configuration Management Informationen bereit stellen, um eine 'IT Service Managementarchitektur' zu beschreiben. Zunächst seien nochmals die Submodelle der OSI-Management-Architekturen genannt, die anfangs eingeführt wurden. Das Informationsmodell dient zur Beschreibung von Management-Objekten, das Organisationsmodell unterstützt Organisationsaspekte, Rollen und Kooperationsformen. Das Kommunikationsmodell beschreibt Kommunikationsvorgänge zu Managementzwecken und das Funktionsmodell strukturiert die Managementfunktionalität, also für das ITSM die Abläufe und Entscheidungen der ITSM Prozesse.

3.3.1 Informationsmodell

Das Informationsmodell (vergleiche Kapitel 2.2) als Herzstück einer Management-Architektur, spezifiziert einen Beschreibungsrahmen für Management-Objekte, wobei nur managementrelevante Parameter einbezogen werden müssen [HAN99]. Im Folgenden wird versucht, ein Informationsmodell direkt aus den bislang vermittelten Informationen zum Configuration Management abzuleiten.

Das zentrale Objekt des Configuration Managements ist zwar genau genommen die CMDB, es soll nun aber die IT-Infrastruktur und die Abbildung ihrer Komponenten in der CMDB im Mittelpunkt stehen.
Die Management-Objekte im Configuration Management nach ITIL und MOF werden als 'CIs' bezeichnet. Ein deutlicher Unterschied zur Definition von Managed-Objects aus dem OSI-Management ist, dass CIs lediglich Attribute und Relationen zugeordnet werden, siehe Tabelle unter 5.4.1. Allerdings gibt es keine Beispiele zu möglichen CI-Methoden. Daraus könnte man zwar vermuten, dass ein relationales Datenmodell verwendet werden soll. Die ITIL macht dazu jedoch keine weiteren Angaben, insofern bleibt das einzusetzende Datenmodell offen.
Die am häufigsten genannten Informationsobjekte sind Hardware, Software und Dokumentationen, wobei 'Hardware' die Netzwerktechnik mit einschließt. Aber auch 'Facilities', Benutzer, SLAs, Incidents, Problems, RFCs, Changes etc. werden als Beispiele für Informationsobjekte genannt (1(c)i). An anderer Stelle wird erwähnt, dass diskutiert wird, ob Telefone und SLAs einbezogen werden sollen. Aus dieser unklaren Beschreibung lässt sich lediglich erahnen, welche Ausmaße eine CI-Erfassung bekommen kann.
Bezüglich der Granularität der Erfassung von CIs, gibt die ITIL als 'rough guide' das Level eines 'independent Change' an (1(c)v). Ohne nähere Erläuterung bleibt an dieser Stelle fraglich, wovon derartige Changes nun unabhängig bleiben. Ansonsten wird nur auf die Detaillierung nach Bedarf hingewiesen. Diese beschränkte Darstellung ist mangels näheren Hinweisen, beispielsweise einem Vergleich verschiedener Verfahren zur Bedarfsermittlung der Granularität, unzufriedenstellend.
Nach 1(c)iiD wird gefordert, sogenannte 'configuration structures' aufzubauen, welche die CIs mit Abhängigkeiten und Position 'beschreiben'. Bringt man den Strukturgedanken in Verbindung mit den geforderten Abhängigkeiten unter den CIs (1(c)ii), so sind für ein Informationsmodell folgende Prinzipien erkennbar. Es sollte Enthaltenseinsbäume geben, welche, von 'Top-Level-CIs' ausgehend, die Aufsplittung auf untergeordnete CIs beinhalten ('Child-CIs' sollten übergeordneten 'Parent-CIs' gehören). Die Forderung von CI-Typen wie beispielsweise 'system software', 'servers', 'routers' legt nahe, eine Strukturierung über eine nach oben abstrahierende Klassenhierarchie aufzunehmen.
Der Vorschlag, eine 'connected-to' oder 'used by'-Abhängigkeit einzuführen, beschreibt möglicherweise die Aufnahme von Kontrollfluss-Pfaden, die letztendlich für die geforderte Impact-Analyse hilfreich wären. Die Gruppierung von CIs zur Vereinfachung von Abhängigkeiten kommt als wei-

teres Prinzip vor, siehe 1(c)iv.
Bei der Forderung nach einer CI-Historie (1(c)vii) bleibt unklar, welche Daten einer Protokollierung bedürfen und wie lange historische Daten gesichert werden müssen.
Weitere Anforderungen, die das Informationsmodell betreffen, sind Life-Cycle-Stati für CIs. Beispielhaft wird angegeben, dass ein Softwarerelease die Stati registered, accepted, installed und withdrawn haben könnte. Hierzu gibt es allerdings keine weiteren Einschränkungen oder Beispiele, welche Rückschlüsse auf andere CI-Stati zulassen würden.

Fazit: Die ungenauen und häufig an Beispielen vermittelten Anforderungen lassen sehr schnell erkennen, dass allein mit den Informationen aus ITIL und MOF keine Konkretisierung eines Informationsmodells stattfinden kann. Für ein Weiterkommen in Richtung eines umsetzbaren, allgemeingültigen Konzepts für ein Informationsmodell (Kapitel 5) ist es nötig, die genannten Anforderungen von ITIL und MOF an das Configuration Management, verursacht durch die angrenzenden IT Service Management Prozesse, grundlegend neu zu betrachten.

3.3.2 Organisationsmodell

Das Organisationsmodell einer Managementarchitektur legt die Akteure, das Rollenspiel und die Grundprinzipien ihrer Kooperation fest [HAN99]. Für menschliche Akteure wird in ITIL und MOF zum Configuration Management die Rolle des Configuration Managers, des Configuration Librarian und die Gruppe des Configuration Management Staffs eingeführt und mit Verantwortlichkeiten beschrieben (1(f)ii). Der Management Staff ist gegenüber dem Configuration Manager weisungsgebunden. Eine weitergehende, detaillierte Beschreibungen gibt es nicht.

Fazit: Bedingt durch den Mangel an weiteren Aussagen über Akteure und deren Beziehungen über die oben genannten, einfachen Zusammenhänge hinaus, erübrigt sich eine eingehende Betrachtung des Organisationsmodells innerhalb des Configuration Managements. Nachdem die Organisation des Configuration Managements unternehmensspezifisch nach Bedarf festgesetzt wird, ist eine nähere Betrachtung möglicher Weisungsrechte und Erfüllungspflichten ebenfalls hinfällig. Prozessübergreifend besteht mit dem MOF-Teammodell ein Organisationsmodell, das keinerlei Hinzufügungen benötigt, die im Rahmen dieser Arbeit von Interesse wären. *Insofern wird in Folge das Organisationsmodell nicht explizit betrachtet.*

3.3.3 Kommunikationsmodell

Das Kommunikationsmodell einer Managementarchitektur legt die Konzepte zum Austausch von Managementinformationen zwischen den Akteuren fest [HAN99]. Nachdem oben zum Organisationsmodell festgestellt wurde, dass es innerhalb des Configuration Managements keine nennenswerte, organisatorische Struktur gibt, die sich allgemeingültig beschreiben ließe, ist auch die Art der Kommunikation zwischen den Akteuren des Configuration Managements nicht betrachtenswert. Selbstverständlich ist, dass zwischen menschlichen Akteuren jegliche Kommunikationsmittel zum Einsatz kommen könnten.

Fazit: Die Betrachtung eines Kommunikationsmodells ist damit innerhalb des Configuration Managements hinfällig. In der Inter-Prozesskommunikation ist es allerdings durchaus möglich, durchgesetzte Kommunikationsmittel für ein Kommunikationsmodell zu empfehlen. So ist es beispielsweise üblich, dass für die Abstimmung zwischen Incident-Management und Problem-Management ein Trouble-Ticketing-Werkzeug mit entsprechenden Übergabepunkten verwendet wird. Für die weitere Betrachtung ist derartiges allerdings nicht relevant, da über den Fokus des Configuration Managements hinaus gehend.

3.3.4 Funktionsmodell

Das Funktionsmodell dient zur Strukturierung der Managementfunktionalität nach Funktionsbereichen. Aus den Aktivitäten des Configuration Managements, die in der Gliederung des Anforderungskatalogs verwendet wurden, ergibt sich eine Strukturierung nach Aufgaben, die letztendlich den Funktionsbereichen innerhalb des Configuration Managements entsprechen. Die Aufgaben sind Planung und Optimierung, Überwachung von Änderungen, Beantwortung von Anfragen anderer ITSM Prozesse, Statusüberwachung und Audits sowie Reportgenerierung.

Fazit: Nachdem ITIL und MOF *keine genaue Beschreibung eines übergreifenden Funktionsmodells* für das Configuration Management enthalten, wird in Kapitel 6 ein Arbeitsprozessmodell vorgeschlagen, welches vor dem Ziel eines anwendbaren Trainingsmodells die Funktionsbereiche strukturiert und in Verbindung setzt.

Zusammenfassend zeigen die Schlussfolgerungen aus der Anwendung der Managementmodelle den Grad der Vollständigkeit der ITIL- und MOF-Beschreibungen. Das geringe Augenmerk auf Organisations- und Kommunikationsmodells in ITIL und MOF kann bestätigt werden, da hierbei nur eine geschäftsspezifische Betrachtung sinnvoll ist, während für das Informationsmodell und Funktionsmodell eine weitergehende Formalisierung zur besseren Anleitung einer Umsetzung erarbeitet werden kann.

3.4 Kritische Faktoren bei der Umsetzung des Configuration Managements

In diesem Kapitel werden aus verschiedenen Quellen die Schwierigkeiten in der Motivation, Planung, Umsetzung und im Betrieb von Configuration Management gesammelt, die mitunter Einfluss auf die Konzeptentwicklung nehmen. Aus Ihnen lassen sich indirekt zusätzliche Anforderungen ableiten, die in den Kapiteln zum Configuration Management in der ITIL/MOF-Grundlagenliteratur (vgl. 3.2.2) nicht vorkommen.

Die eigentlichen Problemfelder in der Konzeptionsphase sind nach Absprache mit ITSM Beratern bedingt durch:

- ungenaue Beschreibungen in ITIL und MOF, welche in IT-Abteilungen zu Verunsicherungen über die Art der Umsetzung führen. Laut [HZB04] ist deshalb ITIL-Literatur in der späteren Implementierungsphase nicht mehr relevant.
- fehlende Standards für eine Umsetzung: ein Standard, wie etwa ein durchgesetztes Informationsmodell für den Aufbau von CMDBs, ist bislang nicht vorhanden. Die Folge ist ein erhöhter Einführungsaufwand wegen individueller Planung und Entwicklung. Im Vordergrund stehen dabei die technischen Aspekte der Konfiguration [Vog02].
- fehlende Automatisierung: im laufenden Betrieb einer CMDB wäre es von Vorteil, wenn - über entsprechende Standards - eine durchgängige Infrastrukturerfassung in heterogenen Umgebungen möglich wäre. Bislang fehlen Tools, welche die Möglichkeiten vorhandener Managementlösungen dafür nutzen, vielfältige Konfigurationsdaten unterschiedlicher CI-Typen automatisiert zu sammeln. Insofern liegt der Fokus in Einführungsprojekten meist auf der Bereitstellung eines Werkzeuges, statt auf der Bereitstellung benötigter Konfigurationsdaten [Vog02].
- Configuration Management unterstützt den Lösungsprozess nur sekundär: zur Problemlösung wird üblicherweise direkt an der IT-Infrastruktur gearbeitet, die - mit etwas mehr Rechercheaufwand - diejenigen Informationen preisgibt, welche auch in einer CMDB gehalten werden können. Nachdem insofern die Notwendigkeit der Nutzung einer CMDB fehlt und Optimierungspotenziale durch den Betrieb des Configuration Managements nur schwer

abzuschätzen sind, erhält der ITSM Prozess nur eine eingeschränkte Unterstützung. Das trifft, je nach Koordinationsbedarf, insbesondere auf den Bereich des Operatings zu, in dem Configuration Management häufig als überflüssiger Verwaltungsaufwand betrachtet wird.

- Die Organisationen erwarten zu viel zu schnell: Zu Beginn bestehen meist grosse Erwartungen, so dass in der Phase der Anforderungsentwicklung überspannte Ziele formuliert werden. Die Empfehlung der schrittweisen Einführung wird nicht berücksichtigt, so dass die Projektkomplexität beziehungsweise die Komplexität der Datenerfassung und der Aufwand der Datenverwaltung leicht außer Kontrolle geraten.

Kritische Faktoren für den laufenden Betrieb werden aus Sicht der ITIL-Autoren in 'ICT Infrastructure Management' [OGC02a] wie folgt beschrieben:

- Minimierung der Differenz zwischen IST und SOLL bei der Aktualisierung der CMDB
- Beschleunigung des Audit-Prozesses
- Reduzierung der Service-Fehler sowie Folgefehler von Changes, erzeugt durch ungenaue CI-Informationen
- erhöhte Geschwindigkeit der Komponenteninstandsetzung
- verbesserte Kundenzufriedenheit über die unterstützten Services
- Reduzierung der Anzahl von CI-Duplikaten
- Reduzierung der Kosten für Wartung, Lizenzen, Software und Hardware
- Verbesserung der durchschnittlichen Wartungszeiten für CMDB-Daten
- bessere Ergebnisse in der Risiko-Analyse durch bessere Asset-Informationen

Vor dem Hintergrund dieser Problemstellungen wird nach dem nächsten Kapitel, in dem ein Beispielszenario für Configuration Management vorgestellt wird, ein Informationsmodell sowie ein Arbeitsprozessmodell vorgestellt werden.

4 Szenario in einem Rechenzentrum

Das Leibniz-Rechenzentrum (LRZ) der Bayerischen Akademie der Wissenschaften vereint folgende Funktionen in einer Institution: Es ist das Rechenzentrum der Hochschulen in München, das Zentrum für technisch-wissenschaftliches Hochleistungsrechnen und verantwortlich für den Ausbau und Betrieb des Münchner Wissenschaftsnetzes (MWN).
Die IT-Infrastruktur des LRZ umfasst eine Vielzahl von möglichen CIs, angefangen von Gebäuden und Rechnerräumen über Netzinfrastrukturen, Systeme von Workstations bis zu Großrechnern, unterschiedlichste Peripheriegeräte, Applikationen und Applikationsnetze, Dienstdefinitionen, Dokumentationen und Leistungsvereinbarungen, technisches Personal bis hin zu Endanwendern.
Um die Konzeptentwicklung für die Umsetzung von Configuration Management mit den Anforderungen aus dem Echtbetrieb abgleichen zu können, wurde für eine Szenarioerhebung der Ausschnitt der LRZ-Schulungskurse zugeteilt. Vorweg sei gesagt, dass es in diesem Szenario einsatzbedingt keine höheren Anforderungen an die Verfügbarkeit gibt, beziehungsweise Changes relativ zeitunkritisch eingebracht werden können. Die Anforderungen, die ITIL und MOF stellen, sind jedoch insbesondere für eine bessere Koordination (Verkürzung der Reaktionszeiten, Ausfallzeiten u.ä.) im 24x7-Stundenbetrieb ausgerichtet. Eine erste, angenommene Vorbedingung sei also der *Bedarf an höherer Verfügbarkeit* im betrachteten Ausschnitt. Des Weiteren findet derzeit im Echtbetrieb des Kurs-Supports eine Koordination über den unmittelbaren Austausch der Administratoren statt. Nach Befragung und Vergleich mit dem Einsatz strukturbildender Koordinationsinstrumente, wie beispielsweise einer CMDB, zeigte sich, dass das bisherige Verfahren bereits optimiert ist und keine Änderungswünsche bestehen. Um eine Motivation für veränderten Koordinationsbedarf zu schaffen, müssen also weitere Vorbedingungen gelten. Zum einen sei angenommen, dass es zukünftig wegen *personellen Veränderungen* oder der *Auslagerung von Arbeiten* einen erhöhten Informationsbedarf über die vorhandene Infrastruktur gibt. Zum anderen sei angenommen, das es zukünftig ein *örtlich verteiltes und zeitlich versetztes Arbeiten* der Administratoren geben soll, welches die Notwendigkeit für eine bessere Dokumentation der Veränderungen an der IT-Infrastruktur schafft. Die genannten Vorbedingungen sind generell ausschlaggebend für die Einleitung eines Projekts 'Configuration Management' [EXP].

Im Folgenden wird das Szenario, zur besseren Nachvollziehbarkeit und zur Vermittlung der Praxisnähe, im Detail beschrieben. Die späteren Beispiele zur Konzeptverdeutlichung beziehen sich lediglich auf Ausschnitte dieser Beschreibung. Die Kursveranstaltungen des LRZ umfassen Anwenderschulungen für Standardsoftware, beispielsweise aus den Bereichen Office-Software, Grafikbearbeitung und Webseitenproduktion, sowie Administratorschulungen, beispielsweise für die Unix-Administration und Netzwerkverwaltung. Für Schulungen auf Windows-PCs stellt das LRZ zwei Kursräume (Nr. 1535 und PEP, 1. Stockwerk) bereit, in denen insgesamt 34 Workstations mit Windows XP für Kursteilnehmer und Kursleiter zur Verfügung stehen. Das Kursprogramm dauert je ein Semester und umfasst 10-15 Veranstaltungen. Vor Beginn eines Semesters wird von den LRZ-Administratoren 'Desktop-Management' in Zusammenarbeit mit den Kursleitern diejenige Software zusammengestellt und konfiguriert, die im Laufe des Semesters auf den Kursrechnern benötigt wird. Dieser Arbeitsschritt ist verhältnismäßig aufwändig. Für die Installation der Kursrechner (wie auch anderer Rechner im LRZ) kommt ein kombiniertes Verfahren, basierend auf den MS-Remote Installation Services (RIS) für die Betriebssysteminstallation und einer selbst entwickelten Installationsroutine für individuell bereitgestellte und benötigte Softwarepakete zum Einsatz.
Der Installationsvorgang läuft im Kurzen wie folgt ab: Zunächst wird von einer Client-Workstation mit PXE-Fähigkeit beim Booten über DHCP ein RIS-Server im Netz abgefragt, an den zur eindeutigen Identifizierung die BIOS-GUID der Workstation übermittelt wird. Der RIS-Server überprüft

im zugeordneten ActiveDirectory, ob unter der GUID bereits ein Rechner eingetragen ist, um Doppelinstallationen zu vermeiden. Für die Installation des Clients wird vom RIS-Server ein Setupprogramm übertragen, das zunächst ein Login für die Anzeige möglicher, installationsbereiter Betriebssystemimages erfordert. Nach Auswahl eines Images beginnt die Silent-Installation auf dem Client. Anhand eigener Modifikationen der Installationsparameter werden nach der Remote-Installation lokale Scripte ausgeführt, welche nach einem Neustart und Autologin ein Share des Fileservers mounten und von dort bereitgestellte Softwarepackages, je nach selektiertem Nutzungsprofil, nach und nach über paketspezifische Installationsscripte installieren. Dabei wird mitprotokolliert, welche Softwarepakete bereits installiert wurden, um bei Abbruch der Installation oder einer Nachinstallation Informationen über weitere benötigte Packages zur Verfügung zu haben. Des Weiteren wird zur Beschränkung des Benutzerzugriffs im ActiveDirectory eine reduzierte Windows-Policy zugeordnet und Desktopanpassungen vorgenommen. Die gesamte Installationsprozedur wird gewöhnlich unbeaufsichtigt durchlaufen und nicht nur bei Kursrechnern verwendet, sondern auch für alle anderen Windows-Desktop-Systeme des LRZ. Nach Abschluss der Installation steht auf einem Kursrechner (im Beispiel unter 4-1 'PC11') eine Windows-Installation mit einem Kursbenutzer (im Beispiel 'Teilnehmer11') zur Verfügung, dessen Homeverzeichnis mit Kursvorlagen sich auf dem Fileserver befindet. Sowohl die lokal installierten Softwarepakete als auch die Kursvorlagen werden kursübergreifend angelegt. Für den Austausch der Benutzer untereinander gibt es ein Transferverzeichnis mit Schreib- und Leserechten für alle Benutzer.
Das Szenario wurde derart vereinfacht, dass wichtige Bestandteile des Configuration Managements gezeigt werden können. Der Netzplan der betrachteten IT-Infrastruktur ist in Abbildung 4-1 dargestellt.

In Ergänzung der oben genannten Erläuterungen, werden nun die bislang unbenannten Serverkomponenten mit Aufgaben und Beziehungen erklärt. Die Serverkomponenten wie auch ein Teil der Netzwerkkomponenten befinden sich im Serverraum des LRZ (4. Stockwerk). Wegen der zentralen Bedeutung für die Zugriffssteuerung werden aus Redundanzgründen zwei ActiveDirectory-Server betrieben, die beide Anfragen verarbeiten und ihr Directory ständig synchronisieren (ActiveDir1 und ActiveDir2). Der dynDNS wird von den ActiveDirectory-Servern benötigt. Aus Sicherheitsgründen erfolgt der Eintrag der IPs von Client-Workstations in den dynDNS über den Linux-DHCP-Server, der für die Rechner im Netz der Kursräume eingetragen ist. Der MWN-DNS fungiert als Secondary NS für den Windows-dynDNS. Auf den Kursrechnern werden bei der Installation Druckertreiber für Netzwerkdrucker eingerichtet. Beim Drucken erfolgt zunächst die Kommunikation mit dem Windows-PrintSrv, der den Printjob an den Linux-PrintSrv weiter gibt, auf dem Accounting je nach Benutzer betrieben wird. Von dort aus wird der Ausdruck auf den selektierten Netzwerkdruckern vorgenommen.
Die meisten Softwarepakete werden für Kursteilnehmer auf den Kursrechnern installiert. Manche Programme werden in Ausnahmefällen auf dem FileSrv gespeichert, wie beispielsweise aus Gründen der vereinfachten Lizenzschlüssel-Eingabe die Statistiksoftware SPSS. Im nächsten Kapitel wird ein Informationsmodell erarbeitet, in dem das Szenario erfasst wird. Der Zugriff auf SPSS wird dabei beispielhaft für eine Abhängigkeitsanalyse in Kapitel 5.6.4.4 verwendet.
Über die Modellierung im Netzplan hinaus werden später noch folgende Elemente (mit gegebenenfalls weitergehender Detaillierung) aufgenommen: Personen und Personengruppen wie Administratoren oder Kursteilnehmer, Administrations-PCs, Stromversorgungseinheiten, wie beispielsweise eine zentrale USV, an die weitere USVs in den Serverracks angeschlossen sind, sowie Gebäude, Stockwerke und Räume.

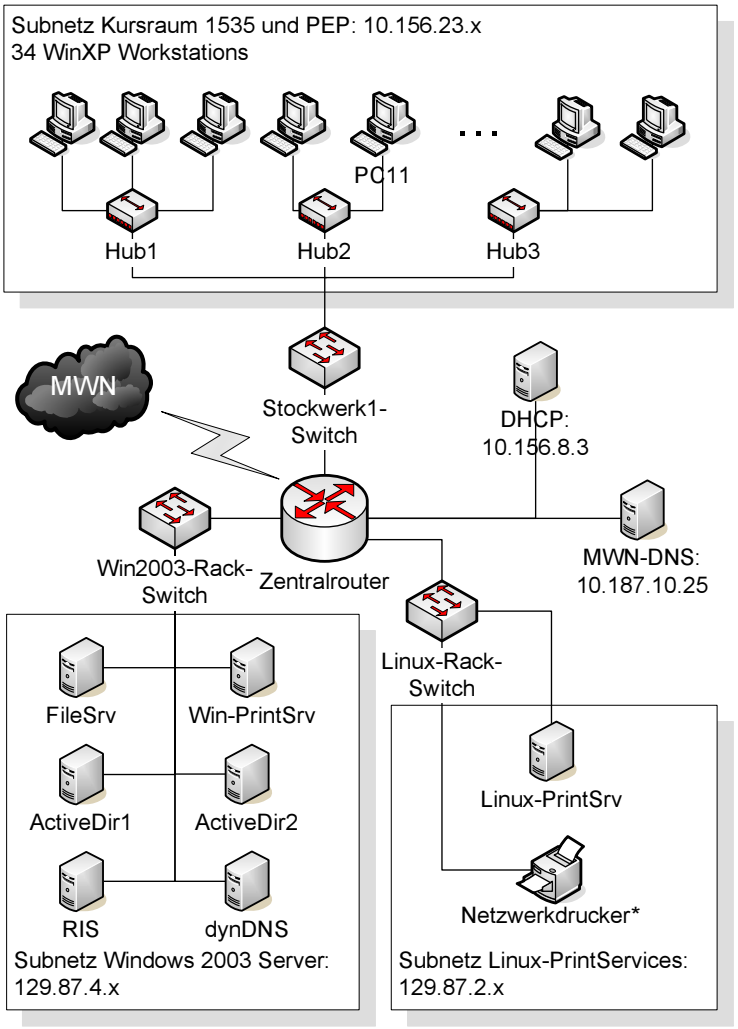


Abbildung 4-1: Netzplan im Szenario

5 Vorschlag eines Informationsmodells nach ITIL und MOF

Bereits einleitend wurde festgehalten, dass die richtige Dimensionierung der CMDB bei der praktischen Umsetzung eine grosse Herausforderung darstellt [HZB04]. Vor dem Hintergrund der Managementproblematik (siehe 2.2) ist dabei fraglich, welche Objekte der IT-Infrastruktur modelliert, welche Attribute zugeordnet und wie die Abhängigkeiten innerhalb des Informationsmodells gestaltet werden sollen.
Wie schon durch die Ergebnisse der Untersuchung von ITIL und MOF nach den GoB (2.6.2, 2.6.3) und durch den Ansatz der Ableitung eines Informationsmodells aus der ITIL (3.3.1) gezeigt wurde, fordern ITIL und MOF für ein Informationsmodell gewisse Eigenschaften, die allerdings nur eine Ausgangsbasis für die Entwicklung darstellen. Dazu gehören beispielsweise die Abhängigkeitsanalyse zur Folgeeinschätzung von Changes und Statusänderungen. Um bei Einführung des Configuration Managements, mit den Zielen des Einsatzes von Referenzmodellen (2-7), Inhalte und die Strukturierung einer CMDB aus einer Gesamtsicht ableiten zu können, ist der Rückgriff auf ein detailliertes Informationsmodells nötig. Selbiges wurde für das Configuration Management nach ITIL und MOF bislang noch nicht veröffentlicht und frei zur Verfügung gestellt [Loo04], weshalb innerhalb dieser Arbeit mit entsprechenden Designentscheidungen ein Vorschlag erarbeitet wurde. Für die bessere Nachvollziehbarkeit des Aufbaus dieses Kapitels sei nochmals auf das Vorgehensmodell unter 1-2 hingewiesen.

5.1 Ausgangslage zur Infrastrukturerfassung

Sammelt man unstrukturiert in einem Brainstorming Infrastrukturelemente und stellt entsprechende Verbindungen her, erhält man eine Graphdarstellung wie in 5-1.

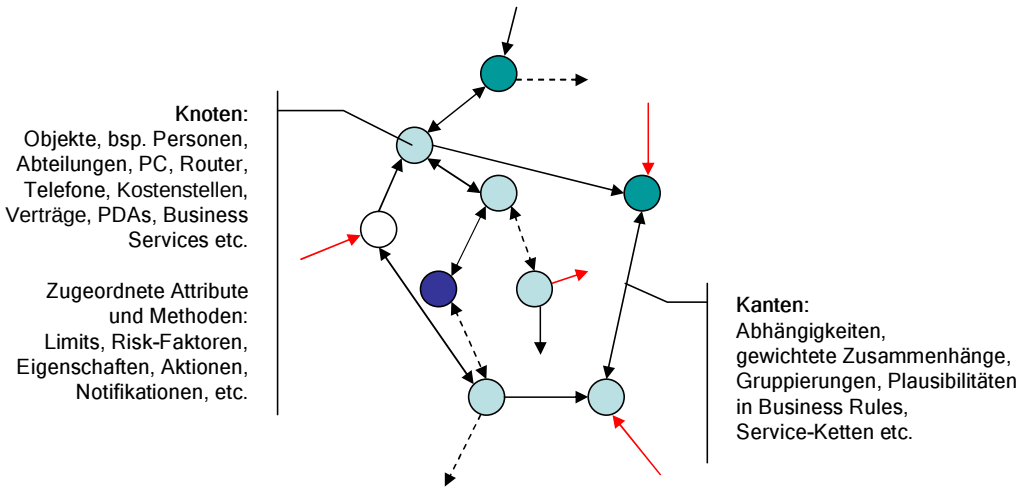


Abbildung 5-1: Konfigurationsgraph, abgeleitet von [Jak04]

Es ergibt sich schnell eine Vielzahl an Objekttypen, angefangen von atomaren Infrastrukturbestandteilen wie Verbindungselemente oder Gebäude, bis hin zu vielfältig, kompliziert zusammengesetzten Elementen wie Business Services, die viele Abhängigkeiten zu anderen Objekten

aufweisen können. Je nach Bedarf und Administrationsfokus werden Attribute von CIs durch Beteiligte unterschiedlich modelliert. Während beispielsweise ein Netzwerkadministrator IP-Adressen zu Ports zuordnet, liegen für Application-Manager IP-zu-Server-Relationen näher. Die Abhängigkeiten werden vielfältig definiert, beispielsweise 'x unterstützt y', 'x hängt ab von y', 'x nutzt y', 'x dokumentiert y', 'x ist ein y' (siehe Vergleichsmodelle unter 5.3).
Als ersten Schritt einer Strukturierung sollen nun die Grenzen der Inhalte für ein Informationsmodell einer CMDB gezogen werden.

5.2 Abgrenzung der Inhalte

Die in Kapitel 3 beschriebenen Mängel von ITIL und MOF bezüglich den Grundsätzen ordnungsmäßiger Modellierung (GoM) bestätigen die Schwierigkeit der Erkennung, Strukturierung und Unterscheidung von Informationsobjekten für ein Informationsmodell. So trifft die ITIL an verschiedenen Stellen wiedersprüchliche Aussagen über die Verteilung von Informationsobjekten in ITSM Prozessen oder enthält nur indirekte Aussagen, welche notwendigerweise einzuführende Informationsobjekte vermuten lassen. Wie schon früher festgestellt wurde, fasst das aus der ITIL abgeleitete MOF die ITIL-Prozessbeschreibungen mit besserer Strukturierung zusammen und erweitert sie um zusätzliche Service Management Functions. Folglich lassen sich daraus in größerem Umfang Informationsobjekte mit Bezug auf das Configuration Management abschätzen, wie im folgenden Kapitel in einer Zusammenfassung einer Auswertung des Autors gezeigt wird. Allerdings reichen auch die MOF-Beschreibungen nicht für eine eindeutige Modellbeschreibung aus, so dass - im Sinne der Ausarbeitung eines Vorschlags - im übernächsten Kapitel eine zweckdienliche Festlegung der Informationsobjekte für die ITSM Prozesse getroffen wird.

5.2.1 SMF-Beziehungen auf das Configuration Management gemäß MOF

Die einzelnen Bände des Microsoft Operations Frameworks [Mic04e] beschreiben jeweils eine der unter 2.5 vorgestellten Service Management Functions (SMF). Sie enthalten in den meisten Fällen, neben einer Beschreibung der Zielobjekte der entsprechenden SMF, die Beziehungen zu anderen SMFs. In Appendix A wurden die entsprechenden Abschnitte, die das Zusammenwirken mit dem Configuration Management beschreiben, aus Gründen der besseren Nachvollziehbarkeit, mit aufgenommen. Eine Zusammenfassung dieser Beschreibungen wurde in Abbildung 5-2 erarbeitet. Von den jeweiligen SMF-Bezeichnungen innerhalb der 4 MOF-Quadranten führen zur Kennzeichnung eines Datenaustausches Verbindungen zur zentralen CMDB, auf denen jeweils die im wesentlichen benötigten Informationen notiert sind (die Pfeilrichtungen haben dabei keine spezielle Bedeutung). Unter den SMF-Bezeichnungen ist beschrieben, für welchen Zweck diese Informationen ausgetauscht werden.

Zwar bieten die ausgetauschten Informationen in der Darstellung und die genannten Verwendungen keine vollständige Beschreibung, lassen aber in der Summe abschätzen, welche CMDB-Inhalte nach MOF gegeben sein sollen. Gleichzeitig wird deutlich, dass in Anfragen an die CMDB relativ häufig der *gesamte Bestand an Configuration Items mit Abhängigkeitsinformationen* benötigt wird, so dass eine differenzierte Betrachtung der CMDB-Inhalte nach Service Management Functions keinen Mehrwert schafft. Der weiter gehende Versuch der detaillierten, Prozess- und SMF-spezifischen Sammlung von Beispielen aus ITIL und MOF, welche abfragespezifisch für gewisse Anwendungsfälle die zugehörigen CIs in einer Datenbank mit Attributen deutlich machen sollten, lässt sich wegen der genannten Mängel in der Referenzmodellierung und insbesondere wegen der geringen Repräsentanz der generell oberflächlich gehaltenen Beispiele nicht fachgerecht durchführen.
Stattdessen muss, nach einer Festlegung der Informationsobjekte im nächsten Kapitel, auf Basis eines anderen Ansatzes festgestellt werden, welche Bestandteile einer IT-Infrastruktur für das serviceorientierte Management im Detail relevant sind.

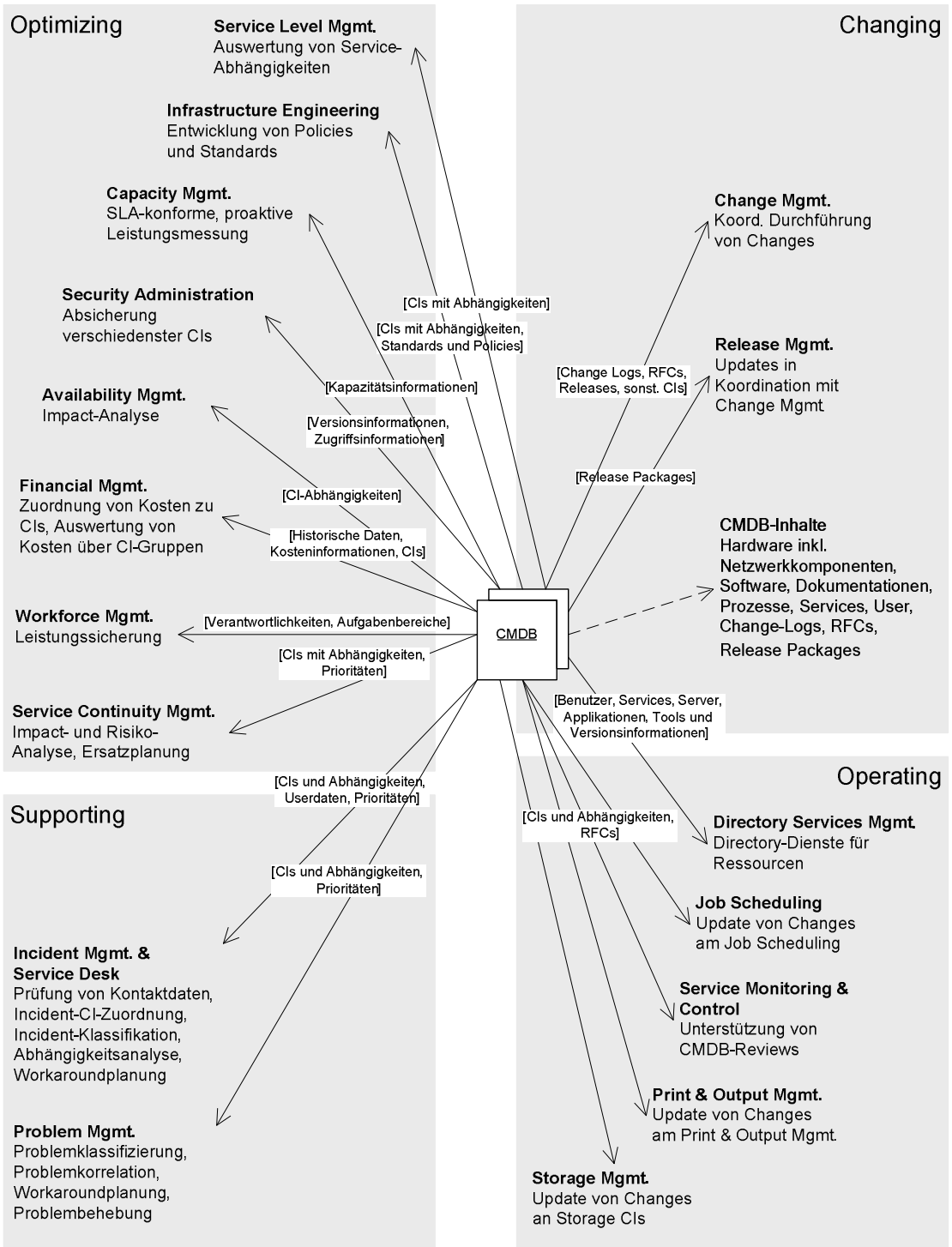


Abbildung 5-2: Zusammenfassung der 'SMF-Relationships to Configuration Management'

5.2.2 Verteilung von Informationsobjekten innerhalb der SMF

In Abbildung 5-2 fällt auf, dass unter den CMDB-Inhalten beispielsweise Change-Logs und Requests for Changes (RFCs) genannt sind, jedoch die in ihrer Art vergleichbaren Incidents oder Problem Records nicht vorkommen. Ursache dafür ist eine unklare MOF-Beschreibung, die nun mit der folgenden Festlegung korrigiert werden soll.
Für die folgenden Service Management Functions sei jeweils die Datenhaltung der Daten im Fokus der SMFs diesen Funktionen selbst zugeschrieben, wobei jeweils Assoziationen zu CIs in der CMDB gesetzt werden können. Die Kardinalität der Relationen entspricht dabei meist 1:n-CIs.

- Incident und Problem Management führen Incidents und Problems in einer eigenen Datenbank mit entsprechenden Verweisen auf betroffene CIs in der CMDB.
- Das Change Management speichert selbst RFCs und Change-Logs (beispielsweise Tätigkeitsbeschreibungen zum Change), ggf. auch Restore-Beschreibungen, mit CI-Verweisen.
- Das Release Management speichert selbst Release Packages mit gegebenenfalls Zuordnungen zu Changes und CIs.
- Das Infrastructure Engineering hält selbst Beschreibungen von Policies und Standards vor und ordnet entsprechende CIs zu.
- Das Service Level Mangement verwaltet Verträge und setzt daraus abgeleitete Limits auf CI-Attribute in der CMDB.
- Das Financial Management übernimmt das Asset-Management und setzt entsprechende Relationen auf CIs in der CMDB (ein CI muss nicht zwingend als Einzelstück in der Vermögensverwaltung erscheinen, insofern ist diese Aufteilung vorteilhaft).

Abzüglich der oben genannten separierten Objekte ergeben sich für die CMDB-Inhalte, hinsichtlich der Aufführung in 5-2, die folgenden IT-Objektklassen zur Ableitung möglicher CIs:

- Hardware mit Netzwerkkomponenten
- Software
- Dokumentationen
- Prozess- und Serviceketten
- Benutzerdaten (inklusive organisationalen Strukturen)

Für SMFs, die sich mit Daten befassen, welche sich 1:1 als Attribute CIs zuschreiben lassen, ist eine Aufteilung auf eine separate Datenhaltung in der oben genannten Form abwegig. So wird festgelegt, dass beispielsweise Kapazitätsinformationen, die in vielfältiger Weise den CMDB-Inhalten zugeordnet werden können (beispielsweise Speicherkapazität bei Hardware, Durchsatz bei Prozesschritten) direkt den entsprechenden CIs zugeordnet werden. Gleiches soll für Prioritäten einzelner CIs oder für die finanzielle Bewertung von CIs zum Zweck der Risiko-Analyse gelten. Ortsinformationen von CIs sollen ebenfalls in die CMDB aufgenommen werden.

Die oben festgelegten Beziehungen sind im Klassendiagramm 5-3 zusammengefasst. Aus Gründen der Übersichtlichkeit und Universalität wurde auf Abhängigkeitskardinalitäten verzichtet. Bezogen auf ein Nicht-CI-Objekt handelt es sich meist um 1:n-CI-Relationen.

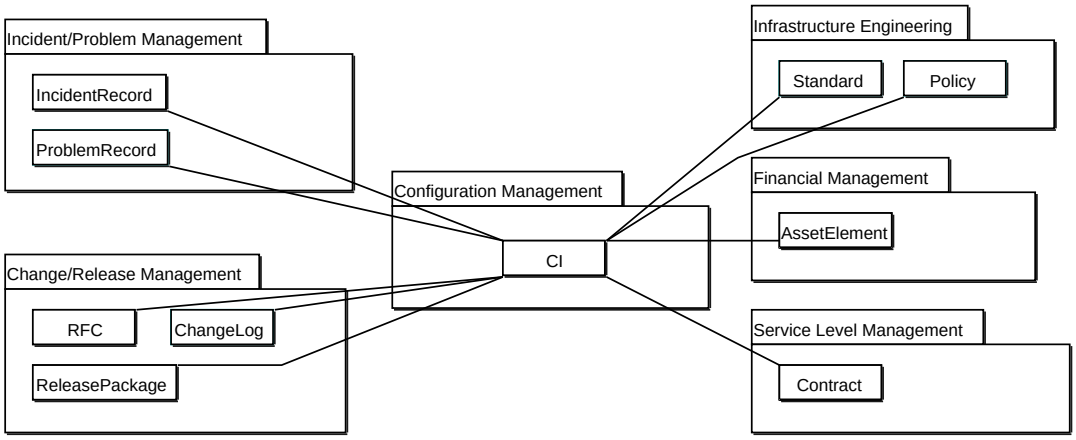


Abbildung 5-3: Klassendiagramm zur Separierung konfigurationsbezogener ITSM-Klassen

5.3 Bestreben nach standardisierter Infrastrukturmodellierung

Nachdem nun eine Festlegung zum Umfang der Objekte für die Erfassung als CIs innerhalb einer CMDB getroffen wurde, geht es nun um die Ausgestaltung des Informationsmodells, also die Auswahl von Objekten, die Zuordnung von Attributen und die Definition von Objektabhängigkeiten. Auffällig bei der Untersuchung von Programmlösungen für CMDBs verschiedener Hersteller [EXP] war, dass diese Lösungen bezüglich der Struktur der zu erfassenden Daten keinerlei vergleichbaren Strukturrahmen vorgaben, der eine Orientierung für eine zweckdienliche IT-Objekterfassung mit Attributen und Abhängigkeiten zuließe. Die individuelle Strukturerfassung in den untersuchten CMDB-Implementierungen geht auf Kundenwünsche zurück, die bestimmt sind durch:

- die Evolution von Konfigurationsmanagement in der Praxis: Zu Beginn erfasste IT-Objekte stellen meist wichtige Bestandteile dar, an denen in Folge erfasste Objekte angehängt werden.
- die Kundensicht, die vielfältig schicht- oder funktionsspezifisch ist. So wird ein Kunde die Verwaltung seiner Rechnerkomponenten zuerst betrachten, während ein anderer Kunde über das Asset-Management an zentraler Stelle die für ihn wichtigen Wirtschaftsgüter der IT-Infrastruktur modelliert.

Zwar ist diese individuelle Modellierung, im Sinne des primären Kundenziels, für kurzfristigen Erfolg die richtige Entscheidung (Erzielen von 'Quick Wins', vgl. ITIL und MOF). Langfristig dürften allerdings die Vorteile einer standardisierten Strukturerfassung überwiegen. Ein standardisierter Strukturrahmen ist aus mehreren Gründen wünschenswert:

- Bessere Kommunikation und Vergleichbarkeit von IT-Infrastrukturen
  - für mehr Transparenz, beispielsweise gegenüber Outsourcing-Partnern.
  - für ein einfacheres Zusammenführen von IT-Infrastrukturen, beispielsweise bei einem Unternehmensmerging oder Konsolidierungsmaßnahmen verschiedener, unternehmenseigener Rechenzentren.
- Bessere Wiederverwendung
  - von Objekt-Templates, die seitens der Hersteller bereitgestellt und bei Aufnahme neuer IT-Objekte lediglich noch mit unternehmensspezifischen Daten gefüllt werden müssten.
  - von Standardsoftware für die IT-Infrastrukturverwaltung. Laut [Mat04] verlangen Kunden von ITSM-Lösungen nach standardisierter Software, statt kostenintensive Individualsoftwareentwicklung in Auftrag zu geben.

- Beschleunigung der Integration: Der standardisierte Strukturrahmen bietet geldwerte Vorteile, nachdem die Diskussion rund um eine eigene Architektur, deren Entwicklung und deren Bereinigung nach Sammlung neuer Betriebserfahrungen reduziert werden.

An dieser Stelle soll ein kurzer Überblick über diejenigen Modelle gegeben werden, die der Entwicklung des vorgeschlagenen Informationsmodells unter anderem zugrunde liegen:

- CIM-Standard, siehe folgendes Kapitel.
- ICIM der Firma Smarts [Sma04]: eine Erweiterung des CIM Standards, unter anderem zum Zweck der Event-Korrelation.
- Metamodell einer CMDB, Firma exagon [EXP]
- Metamodell für Ressourcen aus [Fra04]
- Suntone Architecture Methodology, ein Strukturierungsansatz [SUN01]
- RTE-Architektur aus 'IT-Architekturen strategisch geplant' [NM03]
- Modelling Enterprise IT Architecture, ein ITSM-Ansatz mit ITIL Hintergrund [CT04]

5.3.1 Verwendung des CIM-Standards zur Ausgestaltung

Das Common Information Model (CIM, [DMT04]) ist ein konzeptioneller, implementierungsunabhängiger Standard zur strukturierten Modellierung einer IT-Infrastruktur, der unter der Koordination der Distributed Management Task Force (DMTF) entwickelt wird. Es gibt die Unterteilung von CIM Specification und CIM Schema, wobei hier nur auf letzteres eingegangen wird. Das CIM Schema enthält die eigentlichen, objekt-orientierten Modellbeschreibungen. Vereinfacht dargestellt, ist das Wurzelelement ein 'Managed Element', dessen Subklassen sowohl technologische als auch organisatorische Items beschreiben. Die Klassenhierarchie reicht also von der alles beschreibenden Abstraktion, dem 'Managed Element', bis hin zur Detaillierung unterschiedlichster Elemente. Zur Verdeutlichung des Detaillierungsgrades seien auf unterster Ebene aus unterschiedlichen CIM-Schemaabschnitten die Beispiele EthernetPort, PCIBridge, ServiceIncident, Person, DatabaseSegmentSetting und J2eeURLResource genannt. Ein 'Managed Element' entspricht im Prinzip dem unter 2.2 vorgestellten Managed Object aus dem OSI-Management.
CIM wird von der DMTF als das erste, durch die Industrie gemeinsam verabschiedete und unterstützte Modell zur Beschreibung 'jeglicher Management-Information' bezeichnet. Die Datenstrukturen von CIM wurden seitens verschiedener Hersteller, wie Cisco, Sun, IBM und Microsoft, für die Fortentwicklung und Implementierung eigener Managementlösungen aufgegriffen. Weitergehend wurde der Standard Web-Based Enterprise Management (WBEM) entwickelt, der über HTTP per XML, Operationen auf das CIM-Modell beschreibt. Dieser Standard wurde wiederum von bedeutenden Herstellern aufgegriffen, beispielsweise hat Microsoft WBEM in WMI (Windows Management Instrumentation) für das Desktop-Management seiner Windows-Systeme umgesetzt. Auch in der Unix-Welt gibt es den Einsatz von WBEM, so gibt es zum Beispiel von Sun ein WBEM SDK für Java. Wegen der hohen Verbreitung und herstellerübergreifenden Unterstützung von CIM, wird auf diesen Standard an dieser Stelle stellvertretend eingegangen, um dessen Verwendbarkeit innerhalb eines Informationsmodells für Configuration Management nach ITIL und MOF zu untersuchen.

5.3.2 Problemstellungen in der Objektgranularität und -attributierung

Der Grad der Detaillierung von Objektausprägungen für das Configuration Management kann unterschiedlich ausfallen - ein Rechner mag in einem Fall als 'atomare' Komponente betrachtet werden, in einem anderen Fall mag die Aufführung von Subkomponenten des Rechners und deren Einzelbetrachtung im Sinne der Anwender sein.
In Abbildung 5-4 ist als Anschauungsbeispiel ein Auszug aus dem CIM-Physical-Schema zu sehen.

Der Auszug lässt hinsichtlich der Detaillierung der Objektausprägungen erkennen, welche Problematik besteht, wenn versucht wird, den CIM-Standard durchgängig für die Datenerfassung zu verwenden. Die manuelle Modellierung schon eines Systems ist ein langwieriges und kompliziertes Unterfangen und lässt sich graphisch kaum mehr sinnvoll umsetzen. Nach eigener Auswertung enthält CIM v2.8 insgesamt 1176 Klassen, davon 276 Dependency-Klassen. In der Klassenhierarchie bleiben mangels näherer Begründung Designentscheidungen unklar. So ist beispielsweise nicht ersichtlich, weshalb es eine Unterscheidung zwischen ManagedElements und Components gibt, nachdem sich innerhalb der beiden Vererbungsbäume ähnliche Elemente befinden. Naheliegend ist, dass für die CIM-Modelldesigner aus dem hohen Anspruch der Modellierung 'jeglicher Management-Information' eine zu grosse Anforderung erwachsen ist. Fazit für ein eigenes Informationsmodell ist, dass die *funktionale Reduktion* im Sinne einer Beschränkung auf absolut notwendige Verwendungszwecke des Modells (über die Anforderungen nach ITIL und MOF) eine wichtige Anforderung ist.

Am Beispiel der Erfassung aller NumPhysicalPins eines PhysicalConnectors (5-4) lässt sich vermuten, dass eine derart detaillierte Erfassung für ein Configuration Management, das für eine Gesamtsicht der IT-Infrastruktur aufgebaut wird, am Bedarf vorbei geht. Lediglich die automatisierte Erfassung der Inhalte des Modells, wie beispielsweise über Microsofts WBEM-Umsetzung WMI, ist bei diesem Detaillierungsgrad für eine wirtschaftliche Systeminventur denkbar. Trotz dieses Detaillierungsgrades ist zu beachten, dass sich nicht für jede Konfiguration eine Klassendefinition in CIM finden lässt, also eine individuelle Erweiterung der CIM-Klassen für eine gewünschte Abbildung notwendig ist. Fazit für ein eigenes Informationsmodell ist, dass vorrangig nicht die Festlegung von Objekten und Attributen gefragt ist, sondern *Modellierungsregeln für die flexible Erweiterung und Anpassung* benötigt werden.

5.3.3 Problemstellungen in den Objektabhängigkeiten

Die oben genannten, 276 Dependency-Klassen in CIM v2.8 (beispielsweise CIM_RealizesTapePartition, CIM_USBControllerHasHub, CIM_SAPSAPDependency, CIM_SubProfileRequiresProfile) zeigen, dass in CIM-Abhängigkeiten semantische Information aufgenommen wird, die Funktionalität beschreibt und keine Abhängigkeitsminimierung angestrebt wird. In [Sch03] wird dazu festgestellt, dass die CIM-Schemas eher komplexe Abhängigkeiten aufweisen, die allesamt nachvollzogen werden müssen, bevor sinnvolle Erweiterungen des CIM-Schemas vorgenommen werden können. Gerade durch den gesteigerten Bedarf an individueller Modellerweiterung ist diese Tatsache wegen hohem Einarbeitungsaufwand ein deutliches Negativkriterium für den Einsatz von CIM zur Infrastrukturmodellierung.
Bei der Softwarearchitektur ist neben dem Denken in Schnittstellen und Komponenten die Kontrolle und Vermeidung von Abhängigkeiten die wichtigste Richtschnur [Sie04]. Die Aufnahme einer neuen Abhängigkeit in ein Informationsmodell bedeutet, dass alle Objekte, die diese Abhängigkeit aufweisen dürfen, aktualisiert werden müssen, dass die Anwendungen, die mit dem Informationsmodell arbeiten, um Funktionalität zur Verarbeitung dieser Abhängigkeit erweitert werden müssen und dass seitens der Architekten das Verständnis für die richtige Verwendung dieser Abhängigkeit geschaffen werden muss. Um die Probleme der Komplexität und daraus folgende Unüberschaubarkeit und Ambiguität zu reduzieren, müssen für ein stabiles Informationsmodell möglichst *Abhängigkeiten zwischen Infrastrukturkomponenten minimiert* werden. In den folgenden Ausführungen wird dies unter Beschränkung auf die ITIL und MOF-Anforderungen geschehen.

Fazit: CIM eignet sich wegen seiner Verbreitung und der vorhandenen Toolunterstützung insbesondere zur automatisierten Datenerfassung (beispielsweise über WBEM). Des Weiteren können für die Ausgestaltung eines eigenen Informationsmodells auf die CIM-Klassenhierarchien und Objektattribute zur Orientierung, beziehungsweise für ein Mapping zur Nutzung der automatisierten Datenerfassung, zurück gegriffen werden. Allerdings werden nicht vorrangig ITIL- und MOF-Anforderungen erfüllt - es fällt unter anderem wegen der Modellkomplexität schwer, eine Ab-

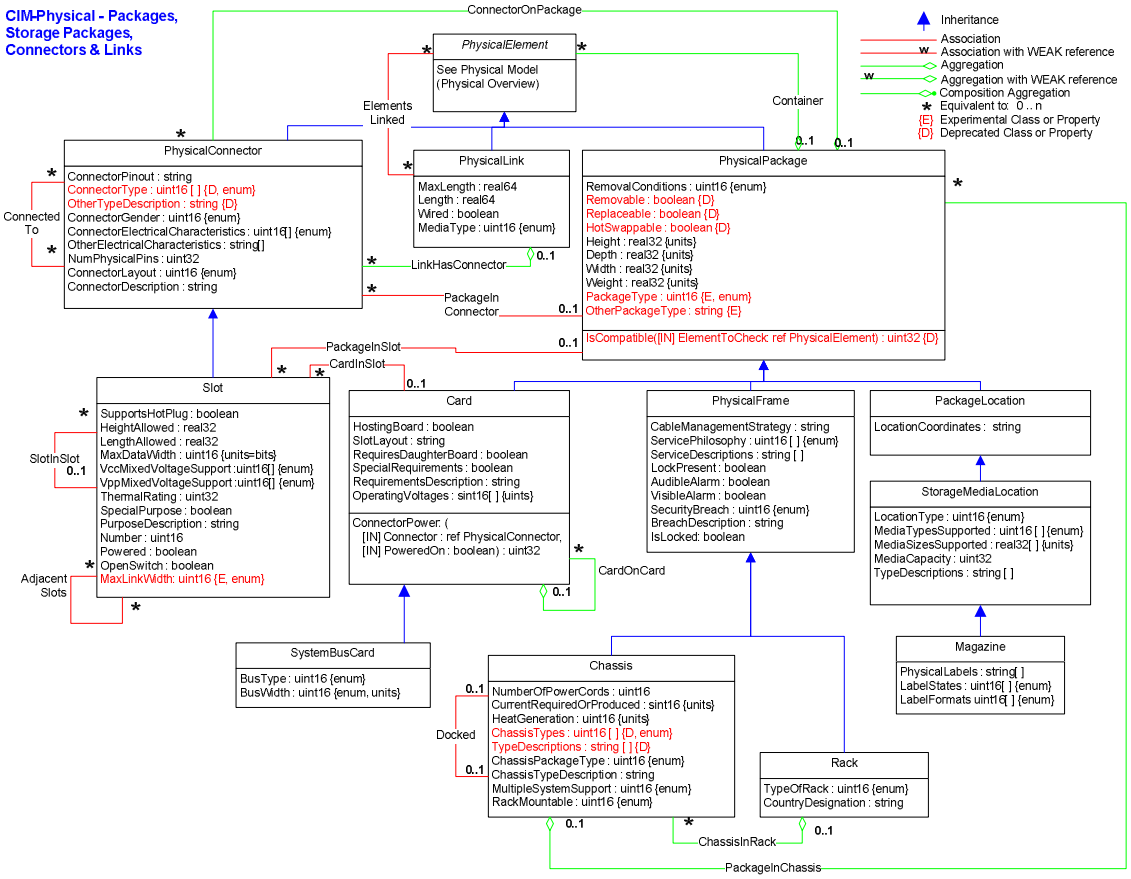


Abbildung 5-4: Beispielhafter Auszug aus dem CIM-Physical-Schema

hängigkeitsanalyse bezüglich Komponentenausfall oder Ausfallwahrscheinlichkeit durchzuführen. Insofern ist der Entwurf eines eigenen, reduzierten Informationsmodells weiterhin notwendig.

5.4 Attributierung von CIs

Nachdem in Kapitel 5.2 eine Festlegung zum Umfang der Objekte für die Erfassung als CIs innerhalb einer CMDB getroffen wurde, geht es nun um die Zuordnung von Attributen zu CIs. Die Schwierigkeit in der Festlegung von Attributen liegt in der Vielfalt der Objekteigenschaften, die fallspezifisch in unterschiedlichem Umfang benötigt werden. Eine Begrenzung möglicher Attribute ist insofern nicht zielführend, stattdessen aber die Erarbeitung allgemeingültiger Attribute und Abhängigkeiten, die sich durch die in ITIL und MOF beschriebenen ITSM-Ziele ergeben. Zunächst soll betrachtet werden, welche Attribute die ITIL beispielhaft für ein CI vorschlägt.

5.4.1 Beispiele aus der ITIL

In ITIL Service Support [OGC00], S. 164, werden CI-Attribute für eine CMDB vorgeschlagen. Sie sind mit einer wortnahen Übersetzung der Beschreibungen in der folgenden Tabelle aufgeführt und werden im Folgenden nach Umstrukturierung als notwendige Attribute betrachtet.

Attribut	Beschreibung
CI Name	Eindeutiger Name des CIs (ITIL schreibt fälschlicherweise 'des CI-Typs').
Copy or Serial Number	Die eindeutige CI-ID, beispielsweise bei Software die Copy Number oder bei Hardware die Seriennummer.
Category	CI-Klassifizierung, beispielsweise Hardware, Software, Dokumentation usw.
Type	Beschreibung des CI-Typs zur Detaillierung der Category, beispielsweise Hardware-Konfiguration, Software-Package, Hardware-Device oder Programmodul.
Model Number (hardware)	Modellnummer des CIs, beispielsweise übereinstimmend mit der Modellnummer des Herstellers.
Warranty expiry date	Ablaufdatum der Herstellergarantie.
Version Number	Die Versionsnummer des CIs.
Location	Die Ortsinformation des CIs, beispielsweise das Medium oder die Bibliothek in der ein Software-CI gespeichert ist, die Adresse oder der Raum, in dem ein Service bereitgestellt wird.
Owner Responsible	Der Name und/oder die Kontaktdaten des CI-Verantwortlichen.
Responsibiliy Date	Datum, an dem die Verantwortung übertragen wurde.
Source/supplier	Die Herkunft eines CIs, beispielsweise 'intern bereitgestellt', 'extern eingekauft'.
Licence	Lizenznummer oder Referenz auf das Lizenzabkommen.
Supply Date	Datum, an dem das CI in der Organisation bereitgestellt wurde.
Accepted Date	Datum, an dem das CI als erfolgreich getestet in der Organisation aufgenommen wurde.
Status (current)	Der derzeitige Status des CIs, beispielsweise 'test', 'live', 'archived'.
Status (scheduled)	Der nächste, geplante CI-Status (mit Datum oder Angabe des Ereignisses, welches den Statuswechsel anzeigt).
Parent CI(s) relationships	Die eindeutigen CI-IDs - name/copy/number/model/number/ der 'parent(s)' dieses CIs.
Child CI(s) relationships	Die eindeutigen CI-IDs aller 'children' des CIs.
Relationships	Alle anderen Beziehungen über 'parent' und 'children' hinaus (beispielsweise CI 'nützt' anderes CI, CI 'ist verbunden' mit anderem CI, CI 'ist enthalten auf' anderem CI, CI 'kann zugreifen' auf anderes CI).
RFC Numbers	Die Schlüssel aller RFCs, die sich auf das CI beziehen.
Change Numbers	Die Schlüssel aller Changes, die sich auf das CI beziehen.
Problem Numbers	Die Schlüssel aller Problems, die sich auf das CI beziehen.
Incident Numbers	Die Schlüssel aller Incidents, die sich auf das CI beziehen.
Comment	Ein Kommentarfeld für freie Einträge, beispielsweise darüber, wie sich die CI-Version von einer Vorgängerversion unterscheidet.

Im obigen Vorschlag ist zwar eine Ähnlichkeit mit einer Tabledefinition einer relationalen Datenbank vorhanden, es gibt allerdings seitens der ITIL keinerlei sonstige Nennung zur Verwendung eines gewissen Datenmodells. Für das eigene Informationsmodell wird sich später aus mehreren Gründen ein objekt-orientiertes Datenmodell anbieten.
Im obigen Vorschlag ist auffällig, dass es keine Anstrengung einer Aufführung gemäß relationaler Entwurfstheorie beziehungsweise die Schaffung einer Normalform gibt. Stattdessen gibt es eine Vermengung von Beispielattributen für verschiedene CI-Typen. Diese Art der Vorschläge passt wiederum zur festgestellten Modellqualität der ITIL nach [HZB04] und legt nahe, dass für eine

detailliertere Modellierung auf einen anderen Ansatz zurück gegriffen werden muss. Im folgenden Kapitel wird eine Annäherung über das ZIFA-Framework beschrieben.

5.4.2 Ableitung über das ZIFA-Framework

Das Zachmann-Framework [Zac80] stellt in einer Matrix Kernfragen über Bezugsobjekte einer Geschäftsmodellierung verschiedenen Detaillierungsebenen dieser Objekte gegenüber. Die Kernfragen, die beispielsweise auch im Qualitätsmanagement für Brainstorming ('sechs Fragen') verwendet werden, schaffen eine ganzheitliche Betrachtung der Business-Objekte:

Was wird betrachtet: Datensicht

Wie findet Interaktion statt: Funktions- beziehungsweise Kontrollsicht

Wo wird agiert: Ortsbezüge, Netzwerksicht

Wer agiert: Personenbezüge, Organisationssicht

Wann wird agiert: Zeitbezüge, auch Reihenfolgenbetrachtung in der Kontrollsicht

Warum wird agiert: Businesstrategien mit abgeleiteten Regeln

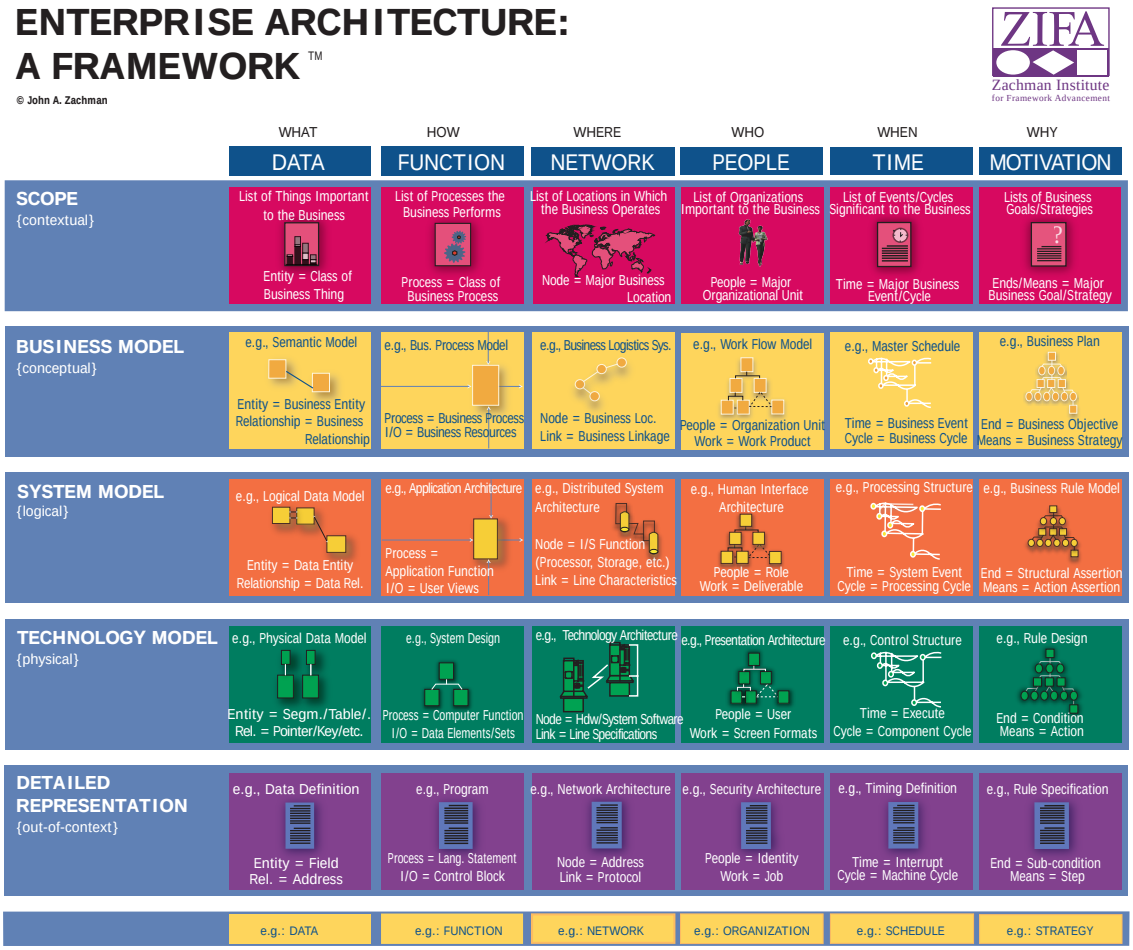


Abbildung 5-5: Zachmann Framework for Enterprise Architecture [Zac80]

Die Ebenenaufteilung verdeutlicht, dass eine Strukturierung von atomaren Bestandteilen (unten) zu vielseitig abhängigen, abstrakten Modelltypen (oben) möglich ist. Später wird gezeigt, dass die hier gewählte Schichttrennung weiter getrieben werden kann, wobei weitergehende Schichttrennungen laufend technologischen Änderungen unterliegen und gerade in höheren Schichten noch wenige Standards etabliert sind.

Für die CMDB-Inhalte ergeben sich nach eigener Festlegung aus den 6 Kernfragen, unter Berücksichtigungen der Anforderungen von ITIL und MOF an das Configuration Management, die folgenden Attributklassen mit Nennung von unterschiedlichen Views der Konfiguration je Klasse. Dieser Katalog wurde mitunter über die Inhalte von Tabelle 5.4.1 entwickelt. Folglich wurden die Attribute und Abhängigkeiten in Attributklassen zusammen gefasst und nach den 6 Fragestellungen geordnet.
Im folgenden Katalog der **Attributklassen** werden in Klammern in kursiver Schreibweise die später gewählten Bezeichnungen im vorgeschlagenen Informationsmodell genannt.

Identifikation (was) Klasse der eindeutigen CI-Adressen für IT-Objekte *(address-Attribut)*, nach ITIL mitunter zusammengesetzt aus hierarchisch übergeordneten Identifikatoren zur besseren, strukturellen Einordnung eines Items *(addressParent-Beziehung)*. Zur Identifikation gehört eine CI-Versionsinformation *(version-Attribut)* und ein Produktbezeichnung (indirekte Verbindung mit der Kategorisierung, siehe unten).
View: Liste aller IT-Objekte
Beispiele:

- Physikalische Labels (Barcodes, RFIDs)
- Seriennummern
- Unique-CMDB-IDs: CI Names
- Modellbezeichnung und Hersteller
- Versionsnummer

Kategorisierung/Klassifizierung (was) Hilfsmittel zur Strukturierung von IT-Objekten über Typisierung, Gruppierung und Generalisierung; Beispielsweise übernimmt eine Klasse 'Change-Manager' die Eigenschaften einer abstrakten Klasse 'IT-Personal' *(hasA- und isA-Beziehungen)*.
View: gemäß einzelner Kategorien, beispielsweise Listing nach Gruppen oder Typen
Beispiele:

- Produkttypen
- Gerätegruppen

Ressourcenbezug (was) Abhängigkeitsinformationen zu anderen IT-Ressourcen, beispielsweise 'ein Server ist abhängig von der Stromversorgung' *(existsOn- und dependsOn-Beziehungen)*.
View: Abhängigkeitsnetz, beispielsweise von Software und Hardware
Beispiele:

- Software hängt existenziell von der Hardware ab, auf der sie läuft
- ein Mailserver hängt von einem DNS-Dienst ab (beispielsweise or-Verknüpfung von Abhängigkeiten zu DNS-Servern oder Abhängigkeit von einem beliebigen Dienst vom Typ 'DNS')
- System-Komponenten hängen vom Gesamtsystem ab

Doku-Bezug (wie) Verweis auf Dokumentationsquellen *(hasDoc-Beziehung).*
View: beispielsweise gruppierte Lizenzzuordnungen
Beispiele:

- Lizenzinformation
- Systemdokumentation
- Mitarbeiterwissen
- Links auf eine DSL (Definitive Hardware Store)

Workflowbezug (wie) Zuordnung eines IT-Objekts zu Prozessactivities oder Events, allgemeine Ressourcenbezüge, beispielsweise 'Gerät a hat in Workflow b Vorgänger c* und Nachfolger d*' *(existsOn- und dependsOn-Beziehungen).*
View: Prozessicht, Kontrollfluss-Graph
Beispiele:

- Applikation mit Bestellfunktion innerhalb eines e-Procurementprozesses
- Failover auf ein Hot-Standbysystem bei Fehlerevents des Produktivsystems

Stati und Limits (wie) Festhalten gewisser Attribute, wie beispielsweise aktuelle Angaben über die Ressourcennutzung/-auslastung, inklusive Limits, bei deren Überschreitung Folgeaktionen ausgelöst werden sollen.
View: Aufgaben für ITSM Prozesse wie das Availability- oder Change-Management
Beispiele:

- SLA-Limits
- Dienst verfügbar mit Auslastung x, nicht verfügbar, Fehler y
- Wahrscheinlichkeit eines Ausfalls *(probability_non_working)*
- Change, Release geplant, durchgeführt, ... *(lifecycle_state)*
- Priorität des CIs

Ortsbezug (wo) Eine räumliche CI-Einordnung, die gewöhnlich nur für physikalische Items relevant ist. Teils ist sie schwer zu erfassen, beispielsweise bei mobilen Geräten, teils ist sie irrelevant (bei kleiner werdenden Strukturen oder bei immateriellen Gütern).
Views: Netzplan, eventuell Raumansichten und Personenverteilungen
Beispiele:

- Raumnummern
- Racks
- Rechnergehäuse
- Ports für Netzverbindungen

Zugriffsbezug/Rechte (wer) Zugriffsberechtigungen durch andere Ressourcen oder Personen *(hasAccessTo-Beziehung)*
View: Berechtigungsnetz
Beispiele:

- Berechtigungsstufen für CMDB-Zugriff
- Berechtigungen für referenzierte Dienste
- Securitykonfigurationen allgemein

Verantwortlichkeiten/Pflichten (wer) Bezieht sich gewöhnlich auf Personen, kann allerdings auch verwendet werden, um Diensten Pflichten, beispielsweise zur Datenlieferung, zuzuordnen. *(hasResponsible-Beziehung).*
View: Aufgabenbereiche von IT- Managern
Beispiele:

- responsible Owner
- First-Level Supporter
- Managementaufgaben allgemein

Zeitbezug (wann) Ergibt sich über vergangene oder geplante/zukünftige CI-Modifikationen *(valid_since- und valid_until-Attribute)*
View: History über CI-Modifikationen
Beispiele:

- Life-Cycle für CIs
- Warranty expiry date
- Planned verification date
- Trigger, die zeitgesteuert Werte setzen oder Systemmeldungen ausgeben.

Finanzinformation (warum) Zeitbedingte Werte (Kaufpreis, Wert nach Abschreibungsdauer, Mietpreise, Leasinginformationen), leistungsbedingte Werte (Kosten nach Nutzungsintensität oder Nutzungshäufigkeit); Verwendung im Asset-Management *(assetno-Attribut).*
View: vielfältig, sämtliche Finanzinformationen, beispielsweise Anlagespiegel, aktuelle Supportausgaben, aktuelle Nutzungsgebühren und ähnliches für das IT-Controlling.
Beispiele:

- Kaufpreis, Mietpreis
- Leasinginformationen (Leasingdauer, Leasinggebühr, Ablöse)
- Anlagegut, Anlagedauer, Abschreibungsart
- Limits für leistungsbedingte Werte
- Budgets
- IT-Accounting
- Wertschöpfungsanteil *(businessvalue)*

Die Prüfung einer gewissen Vollständigkeit der Erfassung der nach ITIL und MOF notwendigen CI-Attribute ist bei derart vielen Einflussfaktoren, welche die obige Strukturierung beeinflussen können, nur angenähert über einen Bottom-Up-Test einzelner, denkbarer CIs möglich. Aus der Betrachtung von Beispielen ergeben sich für die Bezüge zwischen den festgelegten IT-Objekt- und Attributklassen die Ergebnisse in Abbildung 5-6.

Folgerung aus dieser Gegenüberstellung ist, dass *für den Großteil der Attributklassen gilt, dass sie in jedem CI vorkommen* (es gilt Orthogonalität). Lässt sich diese Eigenschaften auf die Ebene einzelner Attribute herunterbrechen, lassen sich in einem Vererbungsbaum relativ viele Attribute im Wurzelelement festlegen. Das hat zur Folge, dass mit Kenntnis des Wurzelelements auf dem gesamten Datenmodell relativ viele Daten ausgewertet werden können. Für die Finanzinformationen sei nochmals darauf hingewiesen, dass sie - wegen hoher Variabilität - separat verwaltet werden sollen.

Objektklassen / Attributklassen	Hardware und Netzwerkkomponenten	Software	Dokumentationen	Prozess- und Serviceketten	Benutzer/Organisation
Identifikation	✓	✓	✓	✓	✓
Klassifikation	✓	✓	✓	✓	✓
Ressourcenbezüge	✓	✓	✓	✓	✓
Doku-Bezüge	✓	✓	teils	✓	✓
Workflowbezüge	✓	✓	✓	✓	✓
Stati und Limits	✓	✓	✓	✓	✓
Ortsbezüge	✓	×	Phys. Dok.	teils	✓
Zugriffsbezüge	✓	✓	✓	✓	✓
Verantwortlichkeiten	✓	✓	✓	✓	✓
Zeitbezüge	✓	✓	✓	✓	✓
Finanzinformationen	teils	teils	teils	teils	teils

Abbildung 5-6: Attributklassen-Objektklassen-Beziehungen im Informationsmodell

5.5 Abhängigkeitsreduzierung

Im oben gezeigten Konfigurationsgraphen 5-1 kann die Anzahl und die Vielfalt der Knoten beliebig sein. Das hat folgende Ursachen:

- Die Granularität der Knoten (also der IT-Infrastrukturkomponenten) kann unterschiedlich detailliert modelliert beziehungsweise gruppiert oder abstrahiert werden.
- Die Fortentwicklung in der IT bringt immer neue Infrastrukturkomponenten hervor, die neu modelliert werden müssen.
- Wegen der Funktionsvielfalt der Knoten ist eine Reduzierung der Eigenschaften und damit eine Generalisierung im Sinne einer Reduzierung der Knotenvielfalt imgrunde nicht möglich.

Statt nun zu versuchen, die Vielfalt der Knoten durch eine standardisierte Strukturierung (vgl. CIM mit entsprechenden Problemstellungen) in den Griff zu bekommen, kann versucht werden, die *Vielfalt der Abhängigkeiten zwischen den Knoten zu minimieren.* Damit ist es gewährleistet, dass Programme, die auf dem Modell arbeiten, Kanten mit einer gemeinsamen, semantischen Bedeutung verarbeiten können, das heißt, das Modell kann zumindest zuverlässig von Knoten zu Knoten abgelaufen werden (für eine Impact-Analyse, für das Sammeln aggregierter Werte von Subkomponenten, für die Suche zulässiger Zugriffswege und ähnliches).

5.5.1 Kommunikationsorientierung

Informations- und Versorgungsflüsse zwischen IT-Infrastrukturkomponenten stellen nach eigener Ansicht minimale Abhängigkeiten dar, sofern die nach ITIL geforderte Impact-Analyse betrieben werden soll. Sie entsprechen den reell existierenden Abhängigkeiten, beispielsweise der Stromzufuhr für Systeme (Versorgungsfluss) oder dem Datenaustausch über Datennetze, physische Dokumente oder die menschliche Sprache (Informationsflüsse). Funktionale Abhängigkeiten (beispielsweise System 'wird abgesichert' von Administrator) oder organisatorische Abhängigkeiten (Dienst 'gehört zur' Dienstklasse) steuern Informations- und Versorgungsflüsse nur indirekt. Sie können für eine Abhängigkeitsminimierung zunächst ausgeblendet werden.
Informations- und Versorgungsflüsse seien im Folgenden unter dem Stichwort 'Kommunikationsorientierung' zusammengefasst. Nach der Kommunikationstheorie gibt es für Information immer

einen Sender und einen Empfänger. Der Sender soll grundsätzlich derjenigen Seite entsprechen, die eine Kommunikation initiiert beziehungsweise auf die Kommunikation angewiesen ist (Ausgangspunkt von →). Beispiele dafür können wie folgt in zwei Klassen eingeteilt werden. Die Syntax dieser Beispiele ist an die XML Path Language (XPATH, [CD99]) angelehnt:

1. `cmdb//application[id=a]` → `cmdb//application[id=b]`
   a ist Client von Server b bzw. a initiiert die Kommunikation mit b

2. `cmdb//person[id=a]` → `cmdb//service[id=b]`
   Person a kontaktiert Service b

3. `cmdb//mainboard[id=a]` → `cmdb//power_unit[id=b]`
   Mainboard a bezieht seine Stromversorgung von b

4. `cmdb//application[id=a]` → `cmdb//system[id=b]`
   Applikation a läuft auf System b

Bei der Betrachtung von → als kommunikationsorientierte Abhängigkeit lassen sich für die 4 Beispiele verschiedene Abhängigkeitsgrade feststellen, die teils vom Typ des Senders und Empfängers abhängen, teils vom Zweck der Kommunikation. So gilt für Beispiel 3 und Beispiel 4, dass die Sender jeweils den Empfänger benötigen, um grundsätzlich zur Verfügung zu stehen. Diese existenzielle Abhängigkeit des Senders vom Empfänger wird zukünftig als *existsOn-Abhängigkeit* beschrieben.
Für Beispiel 1 und 2 benötigt der Sender zwar den Empfänger. Ist der Empfänger aber nicht verfügbar, steht der Sender dennoch für Anfragen durch andere Komponenten zur Verfügung, auch wenn die Korrektheit der Antworten durch die Nicht-Verfügbarkeit des Empfängers möglicherweise gestört ist. Diese Abhängigkeit soll im Folgenden als *dependsOn-Abhängigkeit* beschrieben werden. Beispiel 1 deutet darüber hinaus an, dass der Grad der Abhängigkeit durch den jeweiligen Sender- und Empfänger-Typ bedingt ist. Ist Applikation a ein Programm, das auf dem Betriebssystem b läuft (das hier der Klasse der Applikationen zugeschrieben sei), so besteht die stärkere existsOn-Abhängigkeit. Beispiel 2 deutet an, dass der Grad der Abhängigkeit vom Zweck der Kommunikation abhängig ist. Handelt es sich bei Person a um einen Teleworker, der (ausschließlich) über den Netzdienst b kommuniziert, so ist er existenzabhängig von b.

Zusammenfassung der beiden Abhängigkeitstypen:

a existsOn b a benötigt zwingend (die korrekte Funktionalität von) b, um zur Verfügung zu stehen. Die existsOn-Abhängigkeiten werden entlang der realen Kommunikationswege modelliert.

a dependsOn b a benötigt b für die vorgesehene Kommunikation, steht aber auch (mit fraglicher, korrekter Funktionalität) zur Verfügung, falls b nicht verfügbar ist. Die depensOn-Abhängigkeiten ermöglichen die Direktadressierung einer benötigten Komponente und stellen im Allgemeinen eine Pfadverkürzung von existsOn-Abhängigkeiten dar.

Später werden beide Abhängigkeitstypen noch durch eine hasAccessTo-Abhängigkeiten beschränkt, siehe 5.6.2.1. existsOn ist die härtere der beiden Abhängigkeiten und es gilt existsOn $\subseteq$ dependsOn. Die Unterscheidung der beiden Abhängigkeitstypen dient letztendlich zur weiteren Minimierung der Abhängigkeiten im Konfigurationsgraphen über die Kommunikationsorientierung mit Beschränkung auf unbedingt notwendige Abhängigkeiten. Die existsOn-Abhängigkeiten sind Grundlage für eine Komplexitätsreduzierung durch Schichtbildung, wie sie im nächsten Kapitel erläutert wird. Die Zweckdienlichkeit der dargestellten Abhängigkeitstypen wird in späteren Beispielen weitergehend verdeutlicht.

5.5.2 Entflechtung der IT-Infrastruktur über Schichtbildung

Vorteil eines Layermodells, wie beispielsweise des ISO-OSI-Schichtenmodells, ist - bei Betrachtung einer Schicht - die Verschattung von Funktionalität angrenzender Schichten. Das ist meist verbunden mit der Flexibilität des Technologiewechsels innerhalb einzelner Schichten, ohne Beeinflussung der angrenzenden Schichten. Speziell im ISO-OSI-Schichtenmodell wird die Kapselung der transportierten Daten zur Herstellung einer klaren Schichttrennung verwendet.
Im Bereich der IT-Infrastrukturerfassung sind Übergänge allerdings fließend, vgl. den unterschiedlichen Hardware-Software-Systemschnitt bei RISC oder CISC Mikroprozessoren, oder die unklare Trennung des Trägersystems 'Rechner' (betreffend das Rechnergehäuse) vom Rechner als Rechensystem. Vielmehr gibt es die Möglichkeit, Komponentenkapselung aufzugreifen, um über Komponenten hierarchische Systeme zu modellieren. Ziel dabei ist es, zur Annäherung an die klar definierten Schnittstellen eines Layermodells, *unmittelbare* Existenzabhängigkeiten (existsOn) von untergeordneten Komponenten im Sinne von höherer Atomarität zu betrachten.
Im weiter unten vorgestellten Infrastrukturmodell wird eine nach oben hin offene Hierarchie gezeigt, bei der atomare, unabhängige Komponenten unten und existenzabhängige Komponenten nach oben hin angeordnet sind. Die hierarchische Anordnung ermöglicht die Definition von zu vererbenden Attributen, wie beispielsweise Verantwortlichkeiten für voneinander abhängige Komponenten, welche die Verwaltung des Hierarchiemodells deutlich vereinfachen.

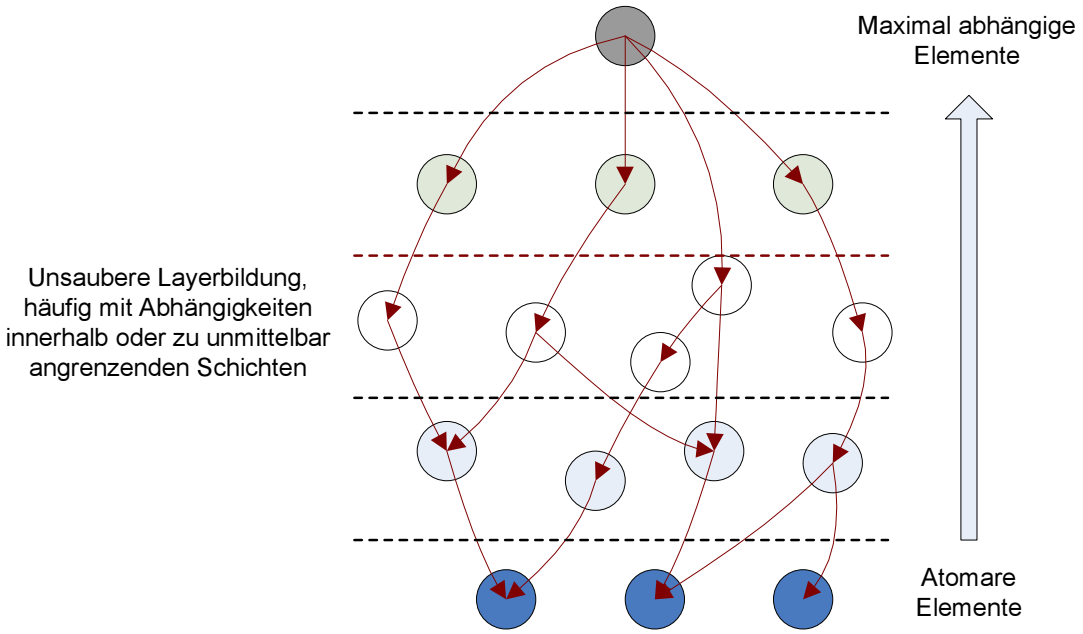


Abbildung 5-7: Ordnungsrelation bzgl. Existenzabhängigkeiten in einer IT-Infrastruktur

Die festgelegten IT-Objektklassen (5.2.2) seien an dieser Stelle mit einer neuen Strukturierung von maximal existenzabhängig bis atomar mit Beispielen genannt. Es werden nun die englischen Bezeichnungen aus dem Vorschlag des Informationsmodells verwendet. Darüber hinaus wurde die Klasse 'Facilities' aufgenommen, welche als atomare Basisklasse für physisch existierende, räumliche Strukturen eingeführt wurde.

IT-User IT-Anwender mit Organisationsstruktur

UsageKnowledge Zusammenfassung von Wissen für die Verwendung der untergeordneten IT-Infrastruktur, wie beispielsweise Benutzerdokumentationen, Verträge, Benutzererfahrung, Business-Prozessdefinitionen, ...

Applications Softwareanwendungen, wie beispielsweise Betriebssysteme, Libraries, Anwendungsprogramme, ...

Systems aktive Komponenten für die Informationsverarbeitung, wie beispielsweise Switches, Workstations, Server, mobile Geräte, Peripheriegeräte, ...

Networks Informations- und Versorgungsnetze mit inaktiven Komponenten, beispielsweise WAN, LAN, Telefonnetze, Stromnetze, ...

Facilities geographische Angaben und physisch existierende Räume, wie beispielsweise Land, Ort, Firmengelände, Gebäude, Stockwerk, Raum, Rack, ...

Diese Klassen entsprechen im Informationsmodell einzelnen Schichten. Innerhalb der einzelnen, teils existenzabhängigen Schichten, kann weitergehend klassifiziert werden (hier gelten allerdings keine allgemeingültig festlegbaren Existenzabhängigkeiten mehr, sondern lediglich individuell zu bestimmende Abhängigkeiten). Abbildung 5-8 zeigt die Entscheidungen zur Einordnung von Komponenten sowie einen Teil der Subklassen, die zur Modellierung des Informationsmodells des Szenarios verwendet wurden.

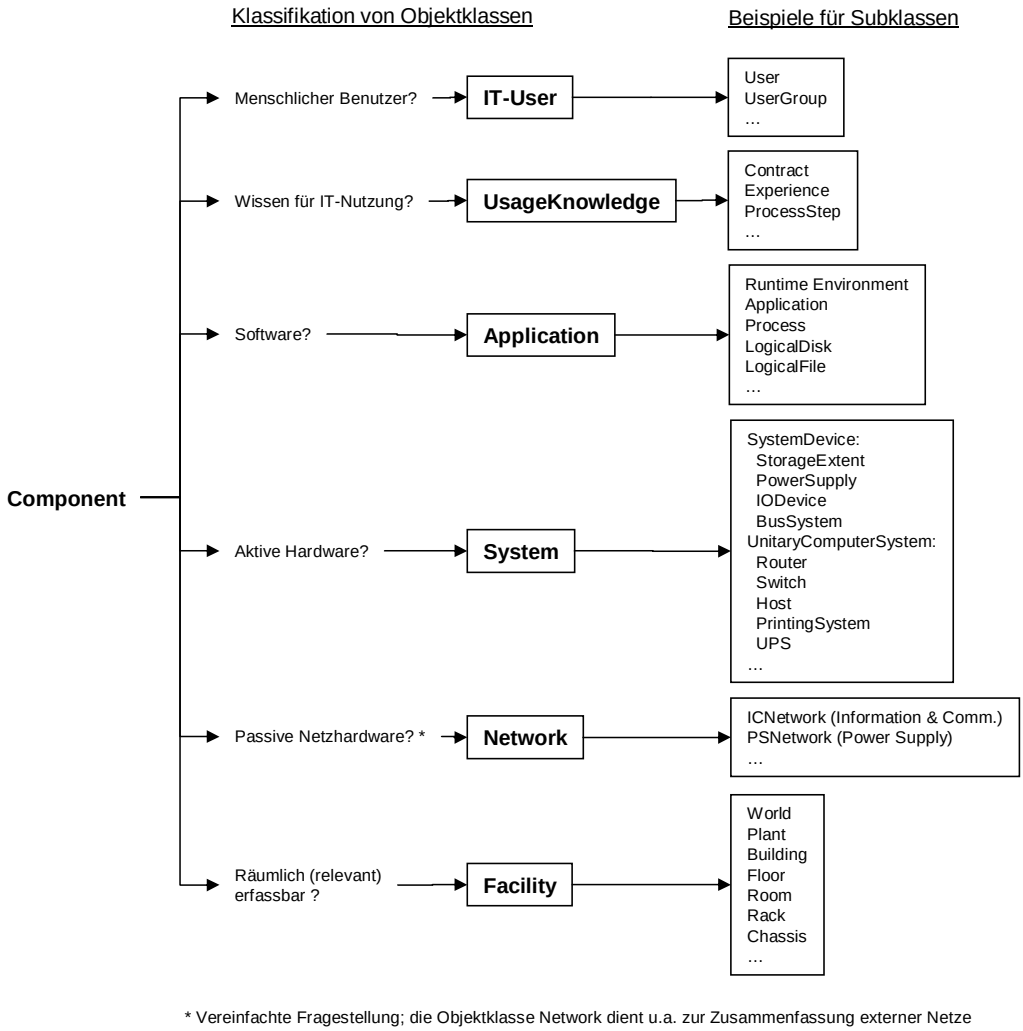


Abbildung 5-8: Komponentenklassifikation und Subklassen

Setzt man nun die oben beschriebenen Klassen beziehungsweise Schichten über *unmittelbare* existsOn-Abhängigkeiten in Bezug, erreicht man eine Schichtbildung wie in Abbildung 5-9.

Diese Abhängigkeiten erklären sich wie folgt: IT-User, die in den User-Layer einzuordnen sind, sind abhängig vom Wissen über die IT-Infrastruktur, das zur IT-Benutzung erforderlich ist (UsageKnowledge). Existiert kein Wissen, kann der IT-User seinen Aufgaben nicht nachgehen. Das Wissen dient dazu, zunächst die physikalisch existierende Systeme im SystemLayer zu benutzen. Softwareapplikationen setzen auf Systeme auf und sind von Systemen existenzabhängig. Die existenzielle Abhängigkeit von IT-Usern oder Wissen auf die Klasse der Applikationen ist individuell gegeben, insofern wurde keine allgemeine existsOn-Abhängigkeit aufgenommen. Ist Software nicht mehr funktionsfähig, können Systeme weiterhin bedient werden - diese Unterscheidung ist letztendlich bedingt durch den physikalisch-logischen Schnitt und die Beschränkung auf unmittelbare existsOn-Abhängigkeiten. Systeme sind wiederum existenzabhängig von Versorgungsnetzen wie dem Stromnetz. Anwendungsbedingt steht beispielsweise ein Terminal nur dann zur Verfügung, wenn ein Datennetz zur Verfügung steht. Insofern wurde eine weitere, allgemeine Existenzabhängigkeit von Systemen auf Netze modelliert. Von der Existenz der zugrunde liegenden, räumlichen Strukturen, sind wiederum alle übergeordneten Klassen abhängig.

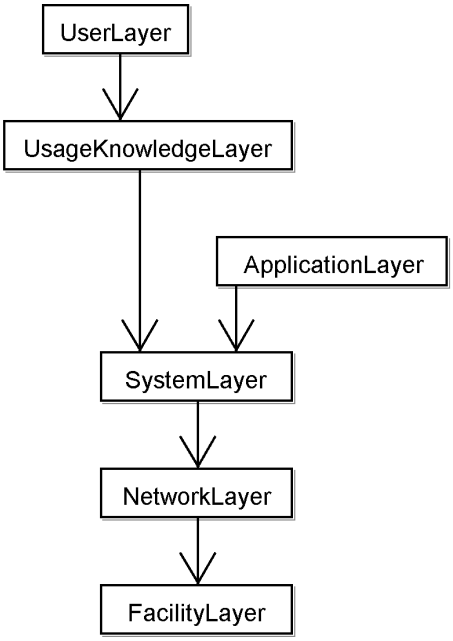


Abbildung 5-9: Layermodell mit unmittelbaren, existenziellen Abhängigkeiten

Zum Vergleich mit anderen Modellen sei die Aussage aus der Veranstaltung [Jak04] genannt, nach der in Schichtenmodellen für CMDBs meist nicht mehr als 5 Schichten, mitunter aus Gründen der Übersichtlichkeit, verwendet werden.

Über die getroffenen Abhängigkeiten hinaus, können existsOn- und dependsOn-Abhängigkeiten auch über mehrere Schichten reichen. In der Modellierungspraxis wird beispielsweise das Wissen eines IT-Users nicht separat modelliert, sondern eine direkte Abhängigkeit IT-User → Rechensystem gesetzt. Des Weiteren gibt es entfernte Anwendungen aus dem SoftwareLayer, die nicht auf eigenen Systemen modelliert werden, sondern nur von einem erreichbaren Netz abhängig gemacht werden. Bei der Modellierung des Szenarios waren diese Vereinfachungen dafür sinnvoll, um sich auf notwendige Komponenten beschränken zu können. Diese Beschränkung reduziert natürlich die Aussagekraft bei der Fehleranalyse, sofern nur für die erfassten Komponenten ein Status-Monitoring betrieben wird. Im Beispiel der entfernten Anwendung könnte ein Verbindungsfehler entweder vom Netz, von angeschlossenen, entfernten Systemen oder von der darauf betriebenen Software bedingt sein. Die Entscheidung für die *Granularität der Modellierung hängt* also bei entsprechendem Monitoring mitunter *von der gewünschten Genauigkeit der Fehleranalyse ab.*

Weitere Beispiele von Abhängigkeitshierarchien verschiedener Komponenten nach existsOn- und dependsOn-Abhängigkeiten verdeutlichen die getroffene Anordnung in 5-9:

- Kunde (UserLayer) → SLA (UsageKnowledgeLayer) → IT-Objekt (aus einem der weiteren Layer)
  - Ohne zu steuerndes IT-Objekt ist ein SLA sinnlos, ein SLA benötigt also ein IT-Objekt, auf das es ausgerichtet ist. Nachdem ein SLA auch dann zur Verfügung steht, wenn die eingebundenen IT-Objekte gestört sind, handelt es sich um eine dependsOn-Abhängigkeit. Selbiges trifft für die Beziehung Kunde → SLA zu.
  - Die Betrachtung in verschiedenen Störfällen gibt eine weitere Bestätigung der Abhängigkeitsdefinition: Ist ein IT-Objekt gestört, dann ist auch ein SLA gestört, folglich ist die Kundennutzung gestört. Ist lediglich ein SLA ungültig (bsp. aus juristisch formalen Gründen), ist ein IT-Objekt weiterhin betriebsbereit, die Kundennutzung aber gestört, da die Vertragsgrundlage fehlt.
- User → Dokumentation → System → Netz → Gebäude
  - Ein Benutzer benötigt eine Dokumentation (bzw. allgemein: Wissen) um ein System nutzen zu können: fehlt ihm die Dokumentation und steht er damit für System-spezifische Anfragen nicht zur Verfügung, kann eine existsOn-Abhängigkeit modelliert werden.
  - Eine Störung des Systems berührt zwar die Dokumentation nicht, da diese Störung keine unmittelbare Veränderung der Dokumentation bedeutet, allerdings wird die Nutzung durch den Benutzer gestört. Eine Störung muss insofern über die Abhängigkeitskette weiter propagiert werden.
  - das System ist von einem Netz abhängig, beispielsweise dem Stromnetz (existsOn).
  - um ein Netz bereit zu stellen, bedarf es einem Gebäude zur Unterbringung. Existiert das Gebäude nicht, steht das Gebäudenetz ebenfalls nicht zur Verfügung (existsOn).

Tabelle 5-10 verdeutlicht beispielhaft, wie einzelne Komponenten für Modellkonformität in verschiedenen Layern separiert werden müssen. Um derartige Komponenten möglichst einfach in das Informationsmodell aufzunehmen, wäre es von Vorteil, wenn bei Komponenten-Bereitstellung Templates mitgeliefert werden würden - mit einer detailgenaueren Untergliederung auf Basis möglichst standardisierter Subklassen.

Beispielkomponente / Layer	**Arbeitsplatzrechner**	**Datenträger mit Software und Doku**	**Arbeitsprozess mit IT-Unterstützung**
IT-User			Beteiligte User
UsageKnowledge		Dokumentation	Phys. dokumentierte oder programmgesteuerte Prozessschritte
Application	Software auf dem Arbeitsplatzrechner	Software	Software, bsp. Programm zur Prozesssteuerung
System	Rechnerkomponenten		Beteiligte Hardware
Network	Stromnetz und Ethernet		Beteiligte Netze
Facility	Gehäuse	Datenträger	Beteiligte Räume

Abbildung 5-10: Komponentenseparierung im Layermodell

Als weiteres Merkmal der Abhängigkeitsreduzierung über dependsOn und existsOn sei genannt, dass Objekte im Allgemeinen bezüglich der Kommunikationsorientierung die Semantik der Abhängigkeiten zwischen den Objekten bestimmen. Unterscheidungen wie in anderen Informationsmodellen (5.3) zwischen 'User *benötigt* SLA', 'Applikation *läuft auf* OS', 'Applikation a *nutzt*

Applikation b' können durch direkte, existenzielle Abhängigkeiten (existsOn) und indirekte Abhängigkeiten (dependsOn) vereinfacht werden: 'User *dependsOn* SLA', 'Applikation *existsOn* OS', 'Applikation a *dependsOn* Applikation b'.
Wie schon in Kapitel 5.5.1 beschrieben wurde, sind die dependsOn-Abhängigkeiten gegenüber den unmittelbaren existsOn-Abhängigkeiten nachrangig. Sie bestimmen weitere Komponentenverknüpfungen, meist innerhalb einer Schicht, und stellen im hierarchischen Infrastrukturmodell, basierend auf existsOn-Abhängigkeiten, kürzere Pfade beziehungsweise Abhängigkeiten dar, welche die Adressierung einzelner (Ziel-)Komponenten berücksichtigen. Betrachtet man also die dependsOn-Abhängigkeiten, setzt man voraus, dass die existsOn-Abhängigkeiten gegeben sind (und somit für die Veranschaulichung einer direkten Verbindung vernachlässigt werden können). Beispielsweise wird für den Zweck der Rückverfolgung von Störungen zwischen einer Anwendungssoftware und dessen IT-User eine dependsOn-Abhängigkeit modelliert, welche eine Pfadverkürzung der Abhängigkeitskette Software → System ← IT-User und gleichzeitig eine Adressierung der benötigten Software (unter mehreren möglichen Softwarepaketen) darstellt.
Abbildung 5-11 zeigt visuell an einem vereinfachten Beispiel die Pfadverkürzung und Adressierung durch dependsOn-Abhängigkeiten. Wie eingangs bereits erwähnt wurde, ist ein effektives Management auf höherer Strukturebene (beispielsweise auf Ebene von dependsOn-Abhängigkeiten) nur möglich, wenn die untergeordneten Schichten (abgebildet über existsOn-Abhängigkeiten) effektiv arbeiten [HAN99]. Das Informationsmodell kommt dieser Anforderung also entsprechend nach.

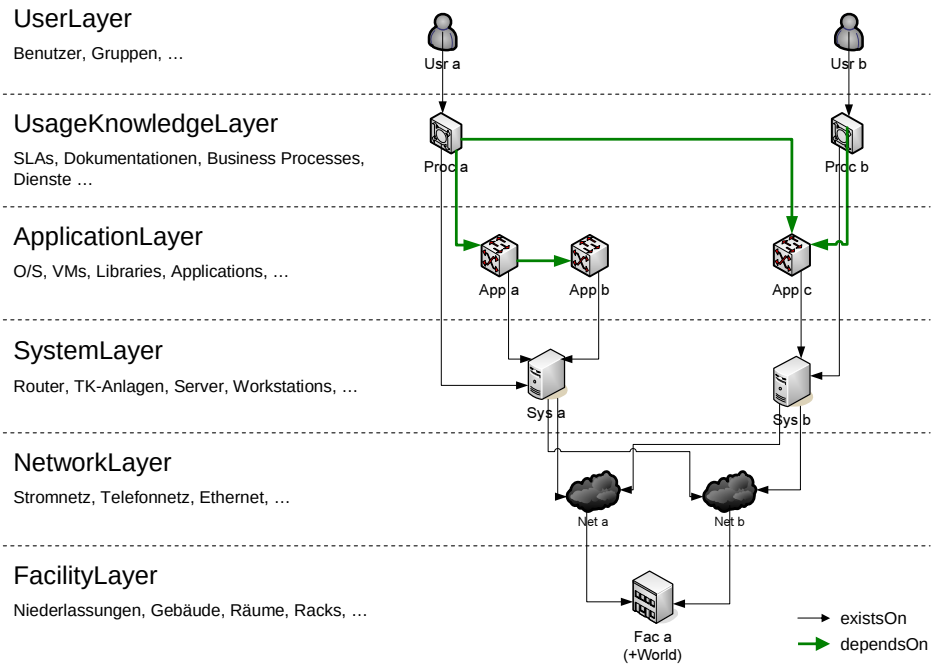


Abbildung 5-11: Beispiel zur Verdeutlichung der existsOn- und dependsOn-Abhängigkeiten

5.5.3 Funktionsausblendung

Auffällig in der Ableitung von Attributen nach dem ZIFA-Framework (5.4.2) ist, dass insbesondere die Aufteilung und Modellierung zu den Fragestellungen 'wie wird agiert' und 'warum wird

agiert' schwer fällt. Hinter dem 'wie' stecken die Funktionalitäten der einzelnen CIs und der Kontrollfluss der Abarbeitung innerhalb der IT-Infrastruktur. Fraglich ist nun, wie Funktionalität verallgemeinert modelliert werden kann beziehungsweise wie viele Entscheidungen berücksichtigt werden müssen, welche die Verzweigung der Kommunikation beeinflussen.
Die Modellierung der Funktionalität von CIs wird in der ITIL nicht gefordert, stattdessen sollen rein textuell die 'funktionalen Spezifikationen' (1(c)i) erfasst werden. Die tatsächlichen, funktionalen Aspekte sind zwar bei der geforderten Abhängigkeitsanalyse hilfreich - wie unten allerdings an einem Beispiel gezeigt werden wird, sind sie nicht zwingend erforderlich. Darüber hinaus ist für IT-Systeme die Vereinfachung der funktionalen Modellierung, wie sie in natürlichen Systemen für Simulationen häufig verwendet wird, problematisch. Nachdem die in der Natur bestehenden Redundanzen für ein immer ähnliches Verhalten der betrachteten Komponenten sorgen und somit auch vereinfachte Betrachtungen zu qualitativ gleichen Ergebnissen führen, ist die funktionale Vereinfachung für natürliche Systeme ein gangbarer Weg zur Komplexitätsreduzierung der Modellierung. In der IT können geringfügige Unterschiede im Entscheidungsablauf das Ergebnis des Gesamtsystems jedoch völlig unterschiedlich ausfallen lassen [All04]. Insofern stellt eine funktionale Vereinfachung innerhalb eines Modells für die Impact-Analyse keine prinzipiellen Vorteile gegenüber einer vollständigen Ausblendung der Funktionalität dar.

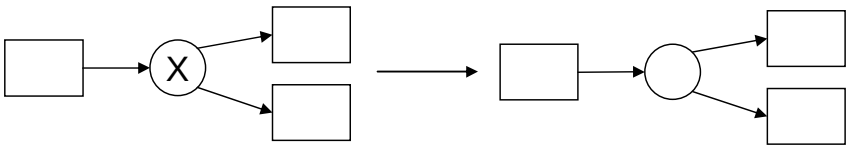


Abbildung 5-12: Funktionale Reduktion

Die funktionale Ausblendung und damit die Reduzierung der Flüsse um Bedingungen (jeder mögliche Output kann damit verwendet werden) reduziert die Komplexität des Modells deutlich. Folge ist allerdings, dass bei einer Impact-Analyse für Fehler in bedingten Prozessabschnitten Gesamtprozesse fehlerhaft gemeldet werden.
Dies soll an einem Beispiel eines Business Prozesses 'Versand von Abrechnungen' gezeigt werden. Für den Versand wird zum einen ein Mailserver, zum anderen ein Briefdruckwerk mit angeschlossener Kuvertiermaschine verwendet. Ist die Kuvertiermaschine defekt, ist bei Reduzierung der Funktion der Gesamtprozess 'Versand von Abrechnungen' gestört, obwohl der Teilprozess für den Mailversand nach wie vor funktioniert. Diese Ungenauigkeit muss im Gegenzug der Vereinfachung in Kauf genommen werden.

5.6 Zusammenführung zu einem Informationsmodell

Bei den folgenden Betrachtungen zum Aufbau eines eigenen Informationsmodells, das den Anforderungen aus ITIL und MOF nachkommt, werden nun die folgenden, vorangegangenen Ergebnisse einbezogen:

- Die festgelegten IT-Objektklassen zur Ableitung möglicher CIs (5.2.2).
- Die Attributklassen zur Attributierung von CIs (5.4.2).
- Die Abhängigkeitsreduzierung über Kommunikationsorientierung (5.5.1), Schichtbildung (5.5.2) und Funktionsreduktion (5.5.3).

5.6.1 Vererbungsbaum für IT-Objekte

Um der Anforderung von ITIL nach einem hierarchischen Aufbau der CIs (1(c)iiB) wie auch dem in Kapitel 5.4.2 herausgearbeiteten Bedarf an Kategorisierung nachzukommen, soll im Informationsmodell ein Vererbungsbaum für IT-Objekte aufgenommen werden. Allgemeingültige Attribute,

wie sie nach ITIL und MOF im nächsten Kapitel vorgeschlagen werden, sollen darin möglichst weitgehend von dem Wurzelelement 'ConfigurationItem' vererbt werden. Zu diesem Zweck wird die *isA-Beziehung* in das Informationsmodell aufgenommen. In Abbildung 5-13 werden die ersten beiden Ebenen eines derartigen Vererbungsbaumes gezeigt. Bereits in Abbildung 5-8 wurden die für die Szenario-Modellierung verwendeten Subklassen dargestellt.

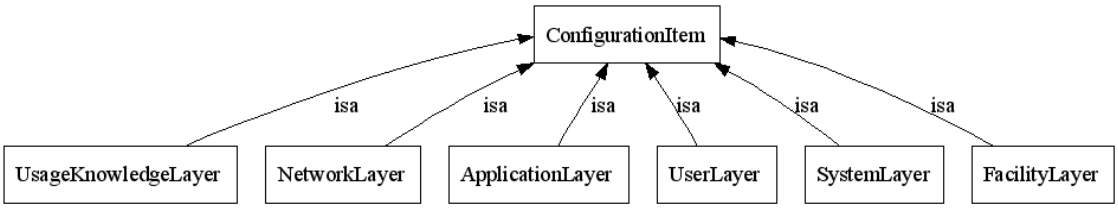


Abbildung 5-13: Oberste Ebenen des Vererbungsbaumes im Informationsmodell (isa)

5.6.2 Generalisierte Attribute nach ITIL und MOF

Für einen konkreten Vorschlag eines Informationsmodells werden nun allgemeingültige Attribute eines Configuration Items und eines Asset Items festgelegt. Asset Items wurden zwar nach Festlegung der IT-Objektklassen (5.2.2) dem Asset-Management beziehungsweise Financial-Management zugeschrieben, nachdem die ITIL innerhalb des Configuration Managements vergleichsweise häufig auf zuzuordnende Attribute eingeht, werden Asset Items im Folgenden dennoch beschrieben.

5.6.2.1 Configuration Item

Attribut address

Beschreibung Zunächst nicht eindeutiger Identifikator (CI Name in 5.4.1) wie beispielsweise App_a, der in Verbindung mit dem rekursiven Ausdruck .addressParent()*.address (vgl. nächste Eigenschaft) zur eindeutigen Adresse wird. Beispiel: host_x.service_y oder datacenter_a.room_b.rack_c.host_d

Typ String

Abhängigkeit addressParent

Beschreibung Referenzierung eines CIs zum Aufbau eines logischen Namens (mit rekursiver Anwendung), siehe address-Attribut. Dieser Abhängigkeitstyp kommt der ITIL-Anforderung nach hierarchischer Namensvergabe nach (2b).

Bezugsklassen managedElement

Kardinalität 0:1

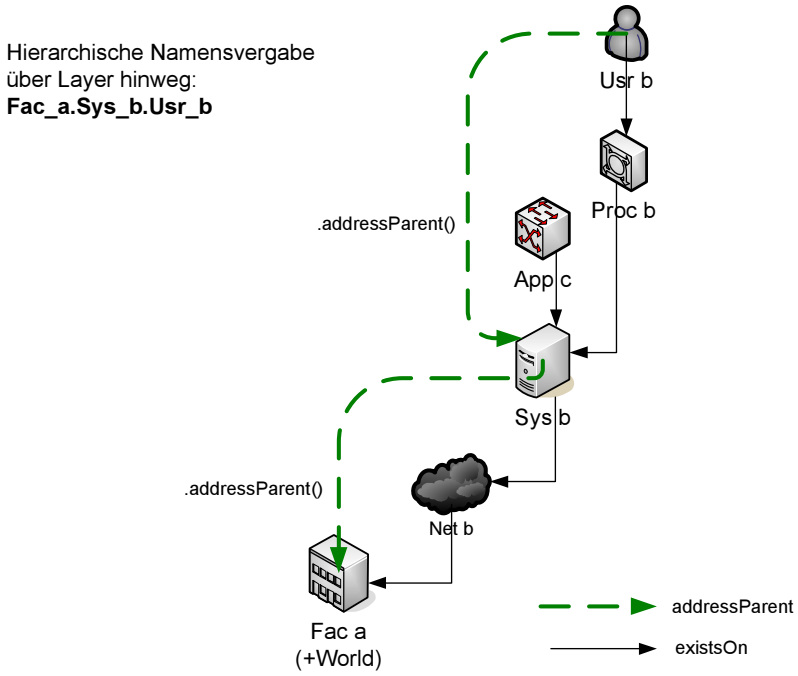


Abbildung 5-14: Hierarchische Namensvergabe (addressParent)

Attribut valid_since

Beschreibung Datum, seitdem das CI in seinem Zustand gültig ist (gefordert in 5.4.1 über diverse Zeitangaben, beispielsweise den CI-Kauf oder die Zuordnung eines CI-Verantwortlichen). Der Zustand ist durch die Summe der Eigenschaften festgelegt. Falls ein Zustandswechsel geplant ist, wird ein Duplikat des CIs angelegt, das im Attribut valid_since das Datum des Wechsels oder des CI-Reviews enthält.

Typ Date

Attribut valid_until

Beschreibung Datum, bis zu dem die Instanz des CIs in seinem Zustand gültig ist. Falls das Ende des Zustands unbekannt ist, bleibt das Feld leer (indirekt gefordert in 5.4.1 über das Feld 'Status (scheduled)')

Typ Date

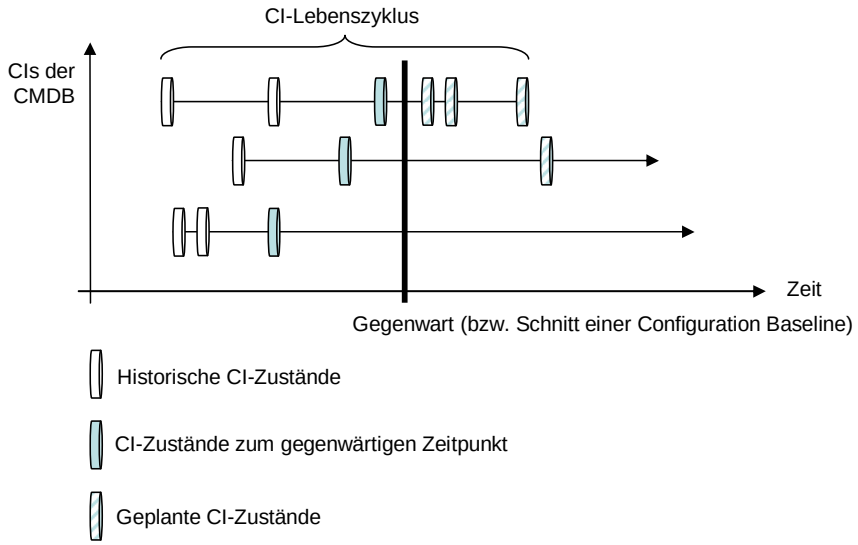


Abbildung 5-15: Folge von CI-Zuständen innerhalb einer CMDB

Attribut lifecycle_state

Beschreibung Der Zustand im Lebenszyklus des CIs (Anforderung unter 1(c)iiiB). In Abbildung 5-16 werden, vorgegriffen auf das Arbeitsprozessmodell in Kapitel 6, denkbare Stati des CI-Lebenszyklus vorgeschlagen. Der Prüfvorgang (inReview), für die Überprüfung der Gültigkeit eines anderen Statusübergangs, wird dabei, entsprechend der ITIL- und MOF- Anforderungen für kontrollierte CMDB-Änderungen, in das Zentrum gesetzt. Bei der Bedienung einer CMDB-Softwarelösung dürfte dieser Zustand jedoch großteils übersprungen werden. Die Aktionen, welche die Übergänge im gezeigten Zustandsgraphen auslösen, sind von denkbar vielfältiger Art - deshalb erfolgt keine explizite Erläuterung. In Abbildung 6-6 werden später Aktivitäten des Configuration Managements für CI-Änderungen mit den genannten CI-Stati in Verbindung gesetzt.

Typ Integer

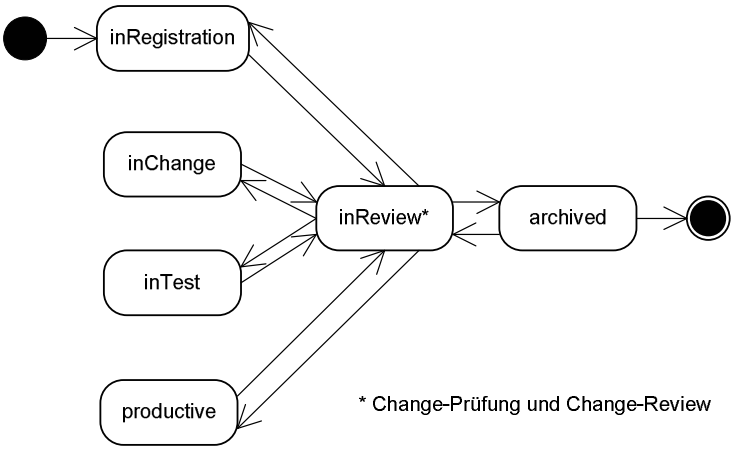


Abbildung 5-16: Vorschlag eines Zustandsübergangsgraphen für CI-Stati

Attribut version

Beschreibung Die Versionsnummer beschreibt die Reihenfolge der Instanzen, beziehungsweise Status-Änderungen eines CIs. Für die Versionsnummer fordert ITIL eine automatische Inkrementierung bei Statuswechsel (ITIL-Anforderung 3(c)ii). Wegen der Überschneidung mit der zeitlichen Information in 'valid_since' könnte das Feld auch weg gelassen werden.

Typ Integer

Attribut probability_non_working

Beschreibung Enthält die Wahrscheinlichkeit eines CI-Ausfalls. Dies ist ein geschätzter Wert (beispielsweise aus Statistiken über eine entsprechende Historie), der allerdings von den sich ständig ändernden Umgebungsverhältnissen abhängig ist und damit nur geringe Aussagekraft besitzt. Falls es untergeordnete Einheiten gibt (hasA-Abhängigkeit), dann ist die Wahrscheinlichkeit das Produkt aller Sub-Wahrscheinlichkeiten (probability_non_working der Sub-Einheiten).
$Risiko = Ausfallwahrscheinlichkeit * Auswirkung$: die Auswirkung wird über existsOn- und dependsOn-Abhängigkeiten in Verbindung mit dem Attribut businessvalue bestimmt, so dass die Berechnung des Ausfallrisikos möglich wird. Die Berechnung des Betriebsrisikos wird nach dem MOF-Risikomodell gefordert (2.5.1).

Typ Float (0-1)

Attribut businessvalue

Beschreibung Das Attribut businessvalue beschreibt den geschätzten, finanziellen Einfluss eines CIs auf die Modellumgebung. Insofern kann dieses Attribut nur für CIs gesetzt sein, die eine Schnittstelle des Modells zur Außenwelt darstellen. Businessvalues können nur für Endpunkte von Workflows (unabhängig auf welcher Schicht des Informationsmodells) festgelegt werden. Im gewöhnlichen wird das Attribut businessvalue nicht gesetzt, sondern indirekt über die Summe der bewerteten CIs an den Endpunkten der existsOn- und dependsOn-Abhängigkeiten bestimmt. Wegen der aufwändigen Feststellung aller Endpunkte und der Schätzung des Wertes, sollte die businessvalue nur qualitative Aussagekraft besitzen. Die Berechnung des Betriebsrisikos, in welche das Attribut businessvalue eingeht, wird nach dem MOF-Risikomodell gefordert (2.5.1). Des Weiteren wird in den ITIL-Anforderungen die Priorisierung von CIs vorgeschlagen (1(c)vi), die beispielsweise über die businessvalue erfolgen kann.

Typ Float

Attribut working_state

Beschreibung Das Attribut enthält die Anzahl der arbeitenden Einheiten. Für logische Einheiten, die aus mehreren Sub-Einheiten bestehen (über eine hasA-Abhängigkeit), enthält working_state die Anzahl der arbeitenden Sub-Einheiten. Eine konkrete Einheit enthält selbst den working_state=1, falls sie korrekt arbeitet. Ist working_state=0, liegt ein Fehler vor. Diese Angaben werden für die nach ITIL und MOF geforderte Impact-Analyse (1(c)iiA) benötigt.

Typ Integer

Attribut assetid

Beschreibung Die Administration der Asset-Informationen wie Kaufdatum, Preis oder Abschreibungsinformationen, wird dem Datenstamm der Finanzabteilung zugerechnet und über die assetid referenziert. Die assetid wird nicht zwingend gesetzt und wird meist für Einheiten (hasA-Abhängigkeiten), wie beispielsweise Computer oder Softwarepakete, eingetragen. Wegen der häufigen Hinweise der ITIL auf Finanzinformationen (vgl. beispielsweise 5.4.1), wird ein Vorschlag eines Asset-Items weiter unten aufgenommen.

Typ Integer

Abhängigkeit existsOn

Beschreibung a.existsOn(b) bedeutet, CI a steht nicht zur Verfügung, solange CI b nicht zur Verfügung steht. Diese Abhängigkeit wird insbesondere für die Impact-Analyse (1(c)iiA) benötigt.

Bezugsklassen CI

Kardinalität 0:*

Abhängigkeit dependsOn

Beschreibung a.dependsOn(b) bedeutet, CI a initiiert eine Verbindung mit CI b (sofern nicht bereits mit existsOn modelliert). Diese Abhängigkeit wird insbesondere für die Impact-Analyse (1(c)iiA) benötigt.

Bezugsklassen CI

Kardinalität 0:*

Abhängigkeit hasAccessTo

Beschreibung Enthält die Zugangsrechte des CIs zur Beschränkung des Informationsflusses. Darüber kann beispielsweise geprüft werden, welche dependsOn-Abhängigkeiten grundsätzlich unter vorhandenen Sicherheitseinstellungen zulässig sind. Im Maximalfall hat jedes CI Zugriff auf ein anderes CI (vollständiger Konfigurationsgraph bezüglich der hasAccessTo-Abhängigkeit). Das bedeutet, dass eine Reduktion der hasAccessTo-Abhängigkeiten gewünscht ist. Es kann zur Reduzierung des Aufwands sinnvoll sein, die Negation von hasAccessTo zu verwalten. Darüber hinaus bietet sich eine Modellierung entlang der reellen Kommunikationspfade (also der existsOn-Abhängigkeiten) an. Wird eine !hasAccessTo-Abhängigkeit (Negation) innerhalb der existsOn-Ordnung auf niedriger Ebene modelliert, wird die Erfassung für höhere Ebenen eingespart. Im Beispiel in Abbildung 5-17 wurde eine !hasAccessTo-Abhängigkeit zwischen Sys a und Sys b eingeführt. Die dadurch unterbundene Kommunikation von Sys a zu Sys b über existsOn-Abhängigkeiten bedingt, dass Proc_a.dependsOn(App_c) ungültig wird.

Bezugsklassen CI

Kardinalität 0:*

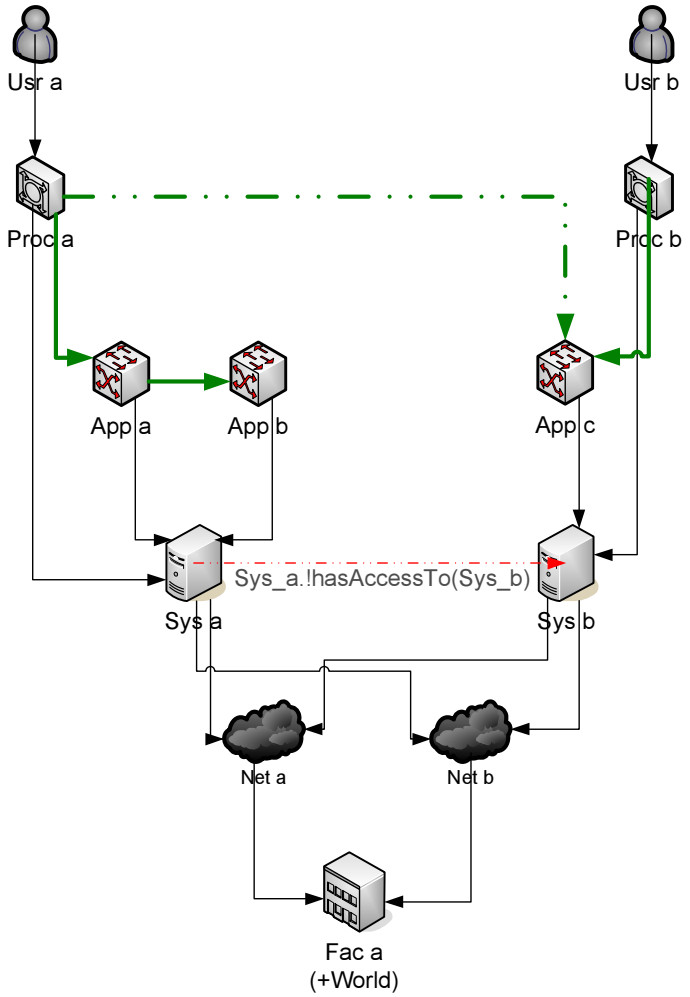


Abbildung 5-17: Beispiel für die hasAccessTo-Anwendung

Abhängigkeit hasA

Beschreibung a.hasA kommt für die Modellierung beziehungsweise Gruppierung von Sub-Einheiten von a zum Einsatz, vgl. Abbildung 5-24.

Bezugsklassen CI

Kardinalität 0:*

Abhängigkeit hasResponsible

Beschreibung Es gibt nach den Anforderungen von ITIL und MOF immer einen verantwortlichen Administrator eines CIs (1(c)iiiA). Für den UserLayer gilt, dass ein Anwender auch selbstverantwortlich sein kann.

Bezugsklassen UserLayer

Kardinalität 1:*

Abhängigkeit hasDoc

Beschreibung Verweis auf Dokumentationen des CIs, sofern verfügbar (1(c)i).

Bezugsklassen UsageKnowledge

Kardinalität 0:*

Attribut Comment

Beschreibung Wird von der ITIL für freiformatierbare Eingaben gefordert, beispielsweise zur Beschreibung der Unterschiede des CI-Status vom Vorgängerstatus, siehe 5.4.1

Typ String

5.6.2.2 Asset Item

Die aufgenommenen Attribute sind abgeleitet aus der Liste der ITIL-Vorschläge 5.4.1.

Attribut assetid

Beschreibung Eindeutiger Schlüssel des Asset Items zur Referenzierung aus der Klasse des Configuration Items.

Typ Integer

Attribut serial

Beschreibung Nach ITIL eine Copy oder Serial Number: Derjenige Identifikator, der ein CI eindeutig identifiziert (in Verbindung mit Herstellerdaten).

Typ String

Attribut modelNumber

Beschreibung Nach ITIL die Modellbezeichnung eines CIs, beispielsweise entsprechend der Herstellerbezeichnung.

Typ String

Attribut warranty_expiry

Beschreibung Nach ITIL das Datum des Ablaufs der Herstellergarantie.

Typ String

Attribut supplier

Beschreibung Angaben zum Hersteller.

Typ String

Attribut licence

Beschreibung Nach ITIL die Lizenznummer (vgl. serial oben) und eine Referenz auf ein Lizenzabkommen.

Typ String

Attribut dateOfAcquistion

Beschreibung Datum des CI-Kaufs.

Typ Date

Attribut priceOfAcquistion

Beschreibung Kaufpreis des CIs.

Typ Float

Attribut depreciationInformation

Beschreibung Abschreibungsinformationen, eher komplex und von vielen Bedingungen abhängig. Diese wie auch weitere Informationen werden von ITIL und MOF nicht näher beschrieben und müssen nach Bedarf vom Finanzmanagement bestimmt werden.

Typ String

5.6.3 Klassendiagramm des Informationsmodells

Um die oben getroffenen Festlegungen in eine Übersicht zu bringen, wurden die Klassen in der Web Ontology Language (OWL, [MH04]) modelliert. OWL baut auf RDF und dem RDF-Schema auf, beides kann in XML und dem XML-Schema beschrieben werden. OWL bietet im Vergleich zu RDF eine reichere Sprache zur Beschreibung von Attributen und Klassen. Unter anderem bietet OWL zusätzliche Abhängigkeiten zwischen Klassen, wie beispielsweise die Disjunktheit, Kardinalitäten, zum Beispiel 'genau ein (Element)', und eine umfangreichere Typisierung von Attributen. Das W3C hat verschiedene Use Cases für den Einsatz von OWL dokumentiert. Darunter befindet sich ein Beispiel für die Klasse der 'Engineering Documentation', in die das Informationsmodell einzuordnen ist. OWL bot sich insofern als eine Variante zur üblichen UML-Objektmodellierung an, die insbesondere wegen der vorhandenen Toolunterstützung ausgewählt wurde. Mit dem Tool Protégé der Stanford University und dem zugehörigen OWL-Plugin ([Knu04]) steht ein vielseitig nutzbares Werkzeug zur OWL-Modellierung zur Verfügung. In der vorliegenden Arbeit wurde das Tool verwendet (5-18), um ein detailliertes Informationsmodell zu modellieren, das über die in Abbildung 5-13 gezeigten Ebenen des Vererbungsbaumes hinaus geht. In der Entwicklung dieses Modells ergaben sich, mit Hinblick auf den Praxiseinsatz und unter Einbezug der ITIL- und MOF-Anforderungen, diejenigen Attribute und Abhängigkeiten, die oben beschrieben wurden.
Um die Übersichtlichkeit zu wahren, wird in Abbildung 5-19 lediglich ein reduzierter Ausschnitt der obersten beiden Ebenen des Informationsmodells, entsprechend dem vorgestellten Vorschlag oben, gezeigt. In der Abbildung werden die gewählten Attribute aus Platzgründen nur auszugsweise dargestellt, wobei anzumerken ist, dass auf zweiter Ebene bereits die Einführung zusätzlicher Attribute erfolgt (beispielsweise in der Klasse UserLayer mit dem Attribut 'Username'). Des Weiteren ist eine Klasse 'managedGroup' zu finden, welche für die Modellierung logischer Gruppen, etwa Redundanzgruppen, verwendet wird.

Kurzreferenz der Abhängigkeitstypen:

a existsOn b a benötigt zwingend (die korrekte Funktionalität von) b, um zur Verfügung zu stehen. Die existsOn-Abhängigkeiten werden entlang der realen Kommunikationswege modelliert.

a dependsOn b a benötigt b für die vorgesehene Kommunikation, steht aber auch (mit fraglicher, korrekter Funktionalität) zur Verfügung, falls b nicht verfügbar ist.

a hasAccessTo b a kann einen Informationsfluss zu b herstellen. Die Abhängigkeit ermöglicht die Modellierung von Zugriffsrechten der Infrastrukturkomponenten und sollte möglichst auf unterer Ebene gesetzt werden.

a hasResponsible b User b ist für Komponente a verantwortlich (nach ITIL und MOF: CI-owner)

a addressParent b b.address ist übergeordneter Namensteil von a.address

a hasDoc b b ist die Dokumentation von a.

a isa b a ist Subklasse von b.

a hasA b a fasst b (und andere Komponenten) zu einer Gruppe zusammen.

Die *-Schreibweise entspricht der üblichen Kardinalitätsangabe 0..infinit.

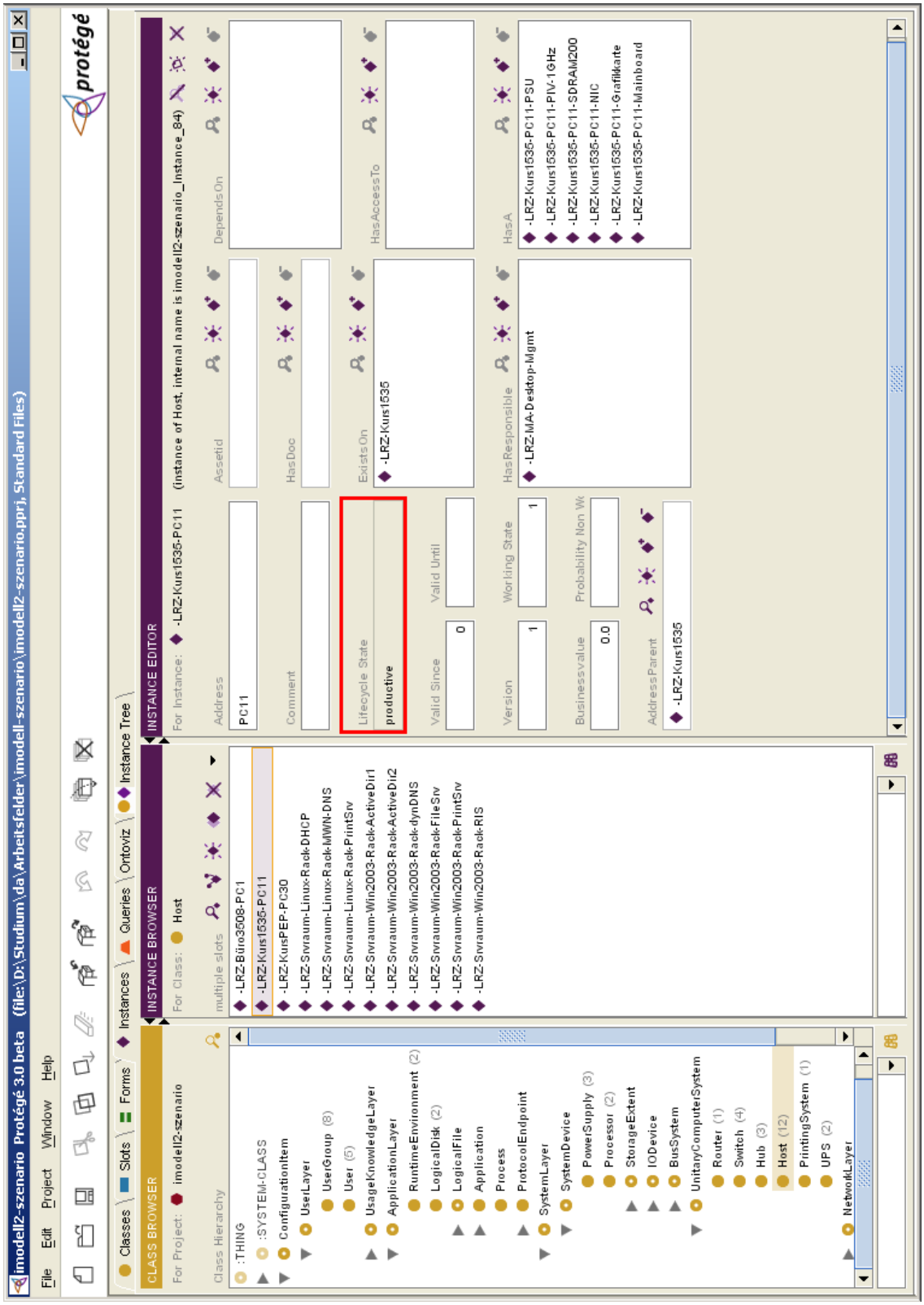


Abbildung 5-18: Design des Informationsmodells im OWL-Tool Protégé

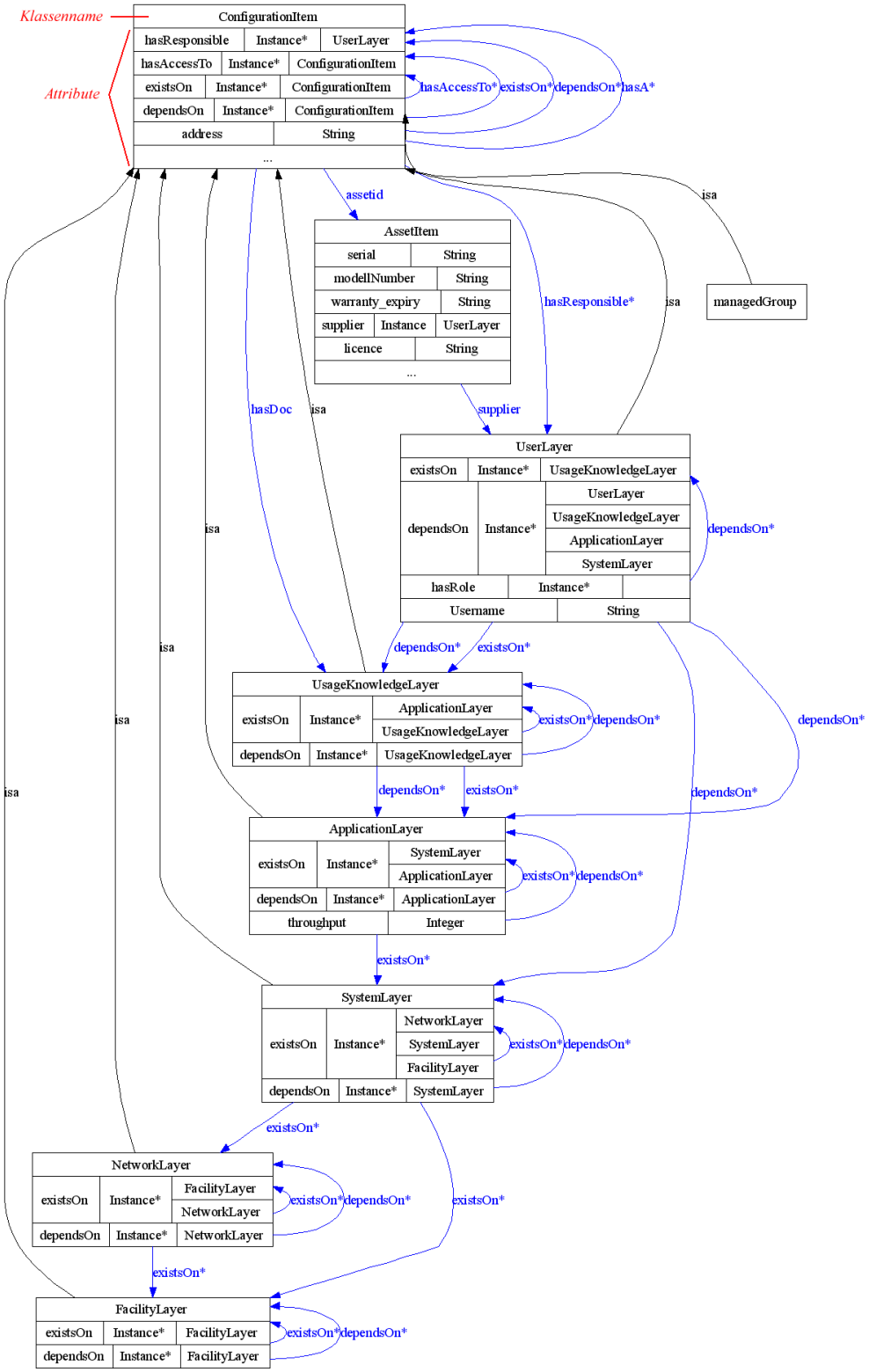


Abbildung 5-19: Vorschlag für Klassenbeziehungen im Informationsmodell

5.6.4 Anwendung auf das Szenario

Zur Evaluierung des Informationsmodells beziehungsweise zur prototypischen Konstruktion einer CMDB, wurde das Informationsmodell mit den Komponenten des unter Kapitel 4 dargestellten Szenarios instanziiert. Die Darstellung kann im Folgenden allerdings wegen der Größe des Modells nur vereinfacht in Auszügen gewisser Modellsichten erfolgen (Schnittbildung). Es werden lediglich ausgewählte Klassen, Instanzen, Attribute und Abhängigkeiten gezeigt. Nachdem das verwendete Visualisierungstool keine manuelle Anordnung der dargestellten Objekte ermöglichte, ist die Ebenenstrukturierung nun lediglich über die Verfolgung der existsOn-Abhängigkeiten nachvollziehbar.

5.6.4.1 Abbildung aus dem UserLayer

In Abbildung 5-20 werden CIs der Klasse 'UserGroup' farblich hinterlegt dargestellt. In der Gruppe 'allPersons' sind alle User zusammengefasst. Die Gruppe 'SPSS-Kurs' enthält alle Kursteilnehmer mit Kursleiter und in der Gruppe der LRZ-Mitarbeiter ('LRZ-MA') gibt es die Untergruppe 'Desktop-Managment', in der zwei Administratoren enthalten sind.

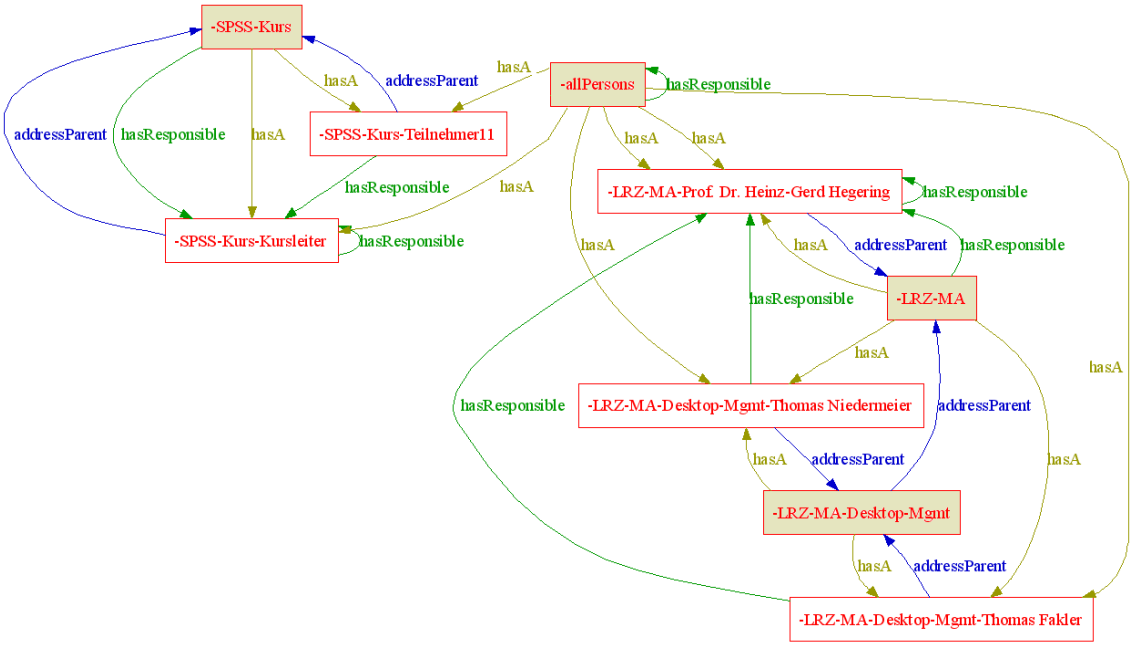


Abbildung 5-20: Auszug von CIs aus dem UserLayer

5.6.4.2 Abbildung aus dem FacilityLayer

In Abbildung 5-21 werden CIs der Klasse 'FacilityLayer' dargestellt (auf 4. Ebene wurden Stockwerke modelliert).

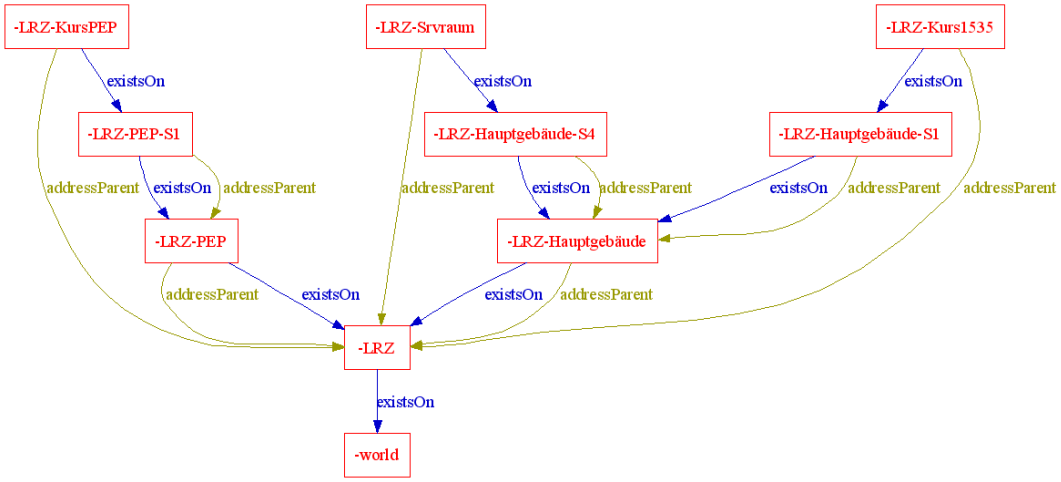


Abbildung 5-21: Auszug von CIs aus dem Facility Layer

5.6.4.3 Layerübergreifende Abbildung

In Abbildung 5-22 werden über verschiedene Ebenen hinweg Komponenten gezeigt, die mit der Nutzung der Software 'SPSS' in Verbindung stehen.

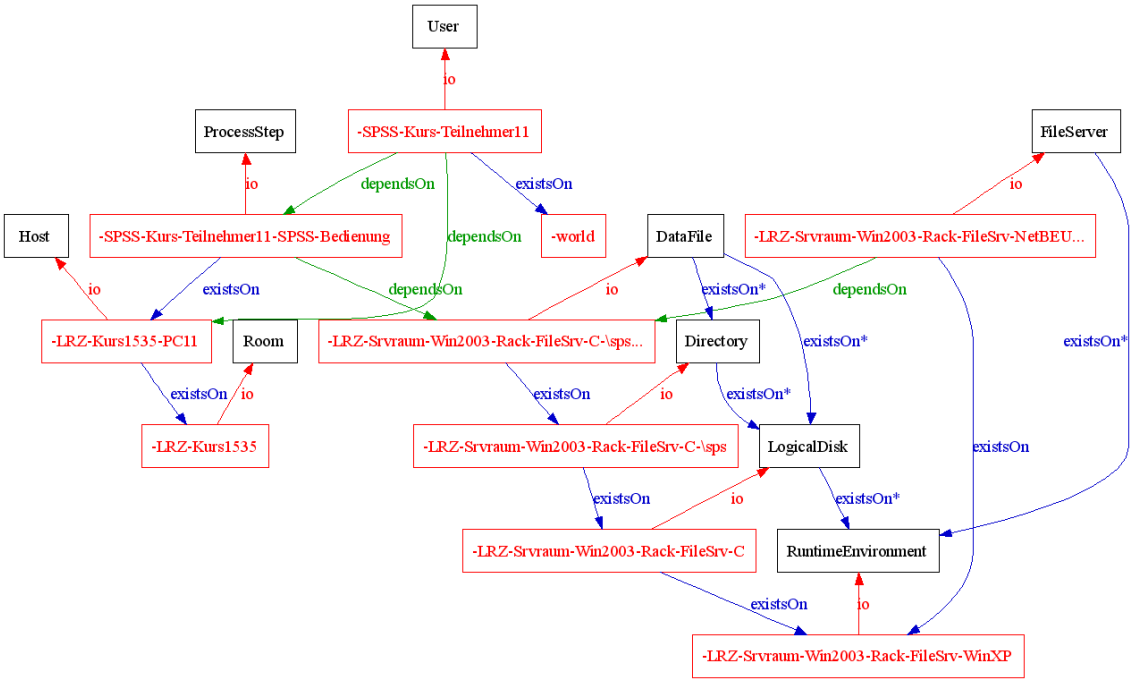


Abbildung 5-22: CIs eines Servicegraphen

Innerhalb der Summe der Prozessschritte unter 'SPSS-Bedienung', kann bei Aufteilung auf mehrere Schritte die logische Abfolge der Bedienung modelliert werden - hierdurch können also rei-

henfolgenbedingte, wechselnde Abhängigkeiten einzelnen Prozessschritten zugeordnet werden. Der Zugriff auf das 'DataFile' *spss**spss.exe* auf dem Fileserver kann nur indirekt über das NetBEUI-Protokoll erfolgen (nicht über den Direktzugriff auf die 'LogicalDisc'). Um diese verkettete Abhängigkeit zu modellieren, können über hasAccessTo die zulässigen Verbindungswege festgelegt werden. Zur Beschränkung der Ausführungen dieser Arbeit wird darauf allerdings nicht näher eingegangen. Einerseits zeigt dieses Beispiel die Möglichkeit der Detaillierung aufgrund der festgelegten Abhängigkeitstypen, andererseits stellt sich unter anderem an diesem Beispiel die Frage der Sinnhaftigkeit einer derart detaillierten Modellierung, vgl. 5.6.6.

5.6.4.4 Elemente des Szenario-Netzplanes

In Abbildung 5-23 wurden Teile des Szenario-Netzplanes 4-1 aufgenommen, welche den vereinfachten Zugriff von 'SPSS-Kurs-Teilnehmer11' über 'LRZ-Kurs1535-PC11' auf benötigte Infrastrukturelemente zeigt. Für die Benutzeranmeldung an PC11 wird mit einem der redundaten ActiveDirectory-Server verbunden. Nur nach erfolgreicher Authentifizierung wird Zugang auf dem 'Win2003-FileSrv' gewährt. An diesem Beispiel wird nochmals deutlich, dass es kurzfristige oder bedingte Abhängigkeiten in der Infrastruktur gibt, die zwar grundsätzlich über Prozessschritte zeitabhängig modelliert werden könnten, die jedoch in der praktischen Anwendung nicht näher betrachtet werden. Stattdessen werden gemäß der funktionalen Reduktion statische Abhängigkeiten aufgenommen, die ein CI bis mindestens zur nächsten überwachten Konfigurationsänderung wechselweise benötigt.

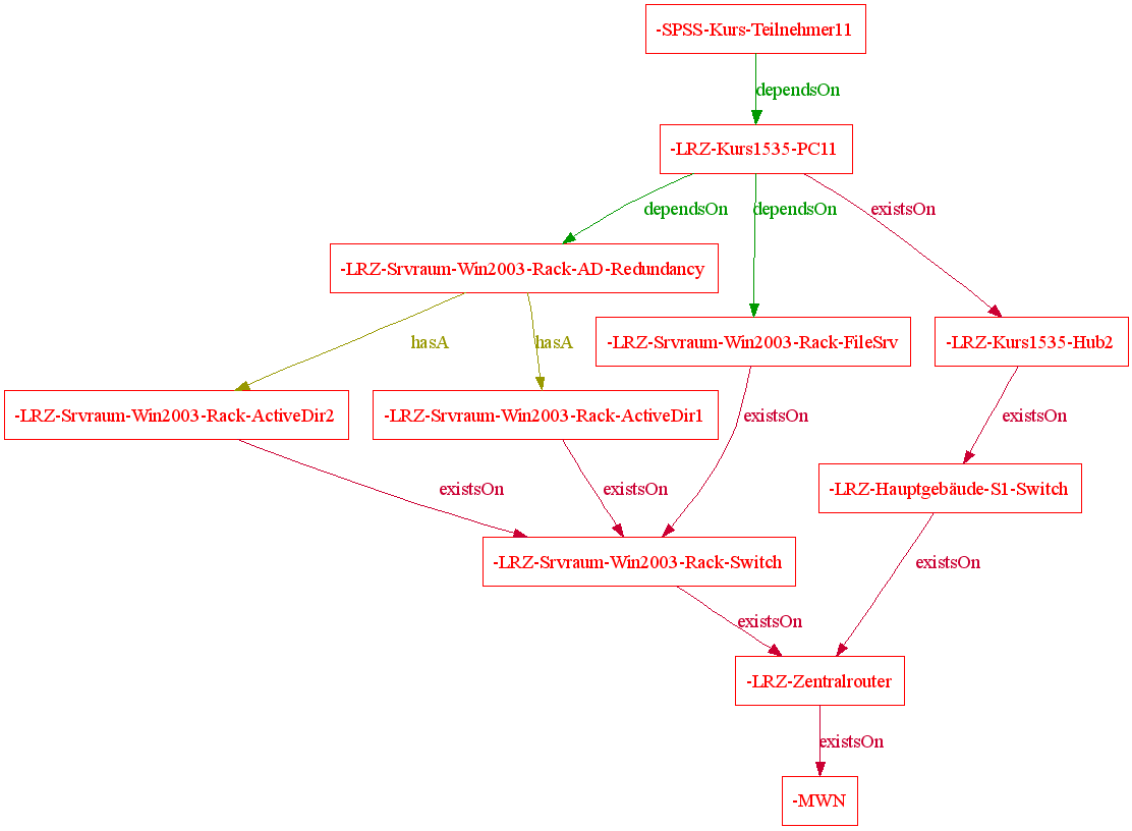


Abbildung 5-23: Teile der Komponenten des Szenario-Netzplanes 4-1

Anhand dieses Beispiels kann der Algorithmus zur Impact-Analyse, der unter der nächsten Überschrift vorgestellt wird, nachvollzogen werden.

5.6.4.5 Abhängigkeitsanalyse im Szenario

An Abbildung 5-23 lässt sich nun eine Abhängigkeits- beziehungsweise Impact-Analyse entlang der dargestellten Abhängigkeiten zeigen. Ein einfacher Algorithmus für die Suche der von einer Störung beeinträchtigten bzw. transitiv abhängigen CIs, wurde in den folgenden Quelltextlistings in der Funktion *propagateErr* beschrieben. Zu berücksichtigen ist, dass dieser Algorithmus lediglich für die Bestimmung der gestörten CIs verwendbar ist, nicht aber für das Auflösen u.U. verschiedener, gleichzeitiger Störungen. Der Quelltext ist zur Selbsterklärung entsprechend kommentiert.

```
public class Dependency {
    public static final int EXISTSON   = 1;
    public static final int DEPENDSON  = 2;
    public static final int HASA       = 3;
    // ... weitere Abhängigkeitstypen
    int type;
    ConfigItem in, out;

    public Dependency(int type, ConfigItem in, ConfigItem out) {
        this.type = type; // Abhängigkeitstyp
        this.out = out;   // ConfigItem am Ausgangspunkt der Abhängigkeit
        this.in = in;     // ConfigItem am Zielpunkt der Abhängigkeit
    }
}
```

```
public class RedundancyGroup extends ConfigItem {

    public RedundancyGroup(String address) {
        this.address = address;
        working_state = 0;
    }

    public void addItem(ConfigItem ci) {
        CIDependencies.add(new Dependency(Dependency.HASA, ci, this));
        working_state++;
    }

}
```

```
import java.lang.*;
import java.util.*;

public class ConfigItem {
    String address;
    int working_state;
    /* weitere CI-Variablen ... */
    List CIDependencies;
    public ConfigItem(String address) {
        this.address = address;
        working_state = 1;
    }

    /* Methoden für die Listenverwaltung in CIDependencies ... */

    synchronized public void propagateErr() {

        // eigenen Funktionszustand herab setzen
        working_state--;

        // falls noch funktionsfähig, dann keinen Fehler weitergeben
        if (working_state>1) return;

        // Fehlerweitergabe über eingehende Dependencies
        for (ListIterator i = CIDependencies.listIterator(); i.hasNext();) {
            Dependency d = (Dependency) i.next();
```

```
27              if (d.out==this) continue; // ausgehende Kanten ignorieren

                if (d.out.working_state == 0)
                    continue; // bereits fehlerhafte CIs ignorieren

32              // Fehler über eingehende Kante weiter geben
                switch (d.type) {
                    case Dependency.DEPENDSON:
                    case Dependency.EXISTSON:
                    case Dependency.HASA:
37                      d.out.propagateErr();
                        break;
                    // case: evtl. weitere Änderungen für weitere Dependencies
                }
            }
42      }
    }
```

Setzt man die Impact- und Risikoanalyse, wie in ITIL und MOF gefordert, als wichtige Aufgabe des Configuration Managements voraus, stellen sie nach eigener Feststellung ein wichtiges Maß zur Bestimmung der Granularität von CIs bereit: CI-Störungen können in ihrer Detaillierung natürlich nur der Granularität der Modellierung entsprechend gemeldet werden. Eine sinnvolle Fehleranalyse ist meist nur dann möglich, wenn verhältnismäßig feingranular modelliert wird.
Um den Begriff der Granularität zu relativieren, sei auf das im Rahmen der LRZ-Befragung [LRZ04] vorgestellte Configuration Management des Leibniz Rechenzentrums verwiesen. In der Softwarelösung 'Remedy ARS' werden hauptsächlich einzelne Systeme mit textuell erfassten Hardwarekomponenten und rechnerspezifischen Angaben wie zugewiesene IP-Adressen, eine textuell erfasste Beschreibung der Systemfunktion und ähnliches verwaltet. Eine Impact-Analyse fehlt in diesem System allerdings, was letztendlich auf die Granularität der Erfassung zurück zu führen ist: Wird ein Dienst beispielsweise über ein dienstspezifisches Monitoring als gestört gemeldet, wird über die direkte Prüfung der IT-Infrastruktur Fehlersuche betrieben. Muss ein Change vorgenommen werden, so werden dessen Auswirkungen von den Administratoren aufgrund der Kenntnis der Infrastruktur abgeschätzt. Die Informationen über die Systeme im genannten Umfang in Remedy ARS, sind für diese Tätigkeit nur wenig hilfreich (beispielsweise nur für den Einstieg zur Fehlersuche oder für die Suche nach Verantwortlichen [Mat04]). Nachdem diese Praxis allgemein üblich ist [EXP] und dies auf eine, nach aktuellen Verhältnissen, Kosten-/Aufwands- vs. Nutzenoptimierung hindeutet, scheint der *Anspruch von ITIL und MOF bezüglich der Bereitstellung von Impact- und Risikoanalyse hoch gegriffen* zu sein. In der Praxis ist die Risiko-Festlegung manuell notwendig, nachdem ein automatisch berechnetes Risiko im Allgemeinen eine zu geringe Aussagekraft aufweist.
Ergänzend zur Impact-Analyse, die wie oben dargestellt die transitiv abhängigen CIs bestimmt, ist eine weitere interessante Anwendung des Informationsmodells die Suche nach Verbindungswegen über kürzeste Wege. Entlang der tatsächlichen und gültigen Kommunikationspfade kann in Richtung der atomaren CIs nach Schnittpunkten für den Aufbau von Verbindungen gesucht werden. Zur Verkürzung der Ausführungen wird darauf allerdings nicht näher eingegangen.

5.6.4.6 Flexibilität der granularen Erfassung

In der Abbildung 5-24 werden die Komponenten des Kursrechners PC11 dargestellt. Die reduzierten Abhängigkeiten reichen auch auf höherer Detaillierungsebene aus, Assoziationen gemäß der ITIL- und MOF-Anforderungen darzustellen. Die einzelnen CIs können weitergehend detailliert oder mit übergeordneten Komponenten zusammengefasst werden, ohne die umliegenden CIs ungültig werden zu lassen. Beispielsweise könnte das modellierte Mainboard mit dem übergeordneten Rechner zusammengefasst werden - zwar mit Informationsverlust bezüglich der Abhängigkeits- und Attributgenauigkeit, jedoch unter Beibehaltung der Gültigkeit sonstiger CIs.
In den bisherigen Ausführungen wurde der Bestimmung und Detaillierung der CI-Attribute nur wenig Aufmerksamkeit geschenkt. Über die in Kapitel 5.6.2 beschriebenen generalisierten Attribu-

te hinaus, wurden im Szenario häufig nur wenige, zusätzliche Eigenschaften aufgenommen. Eine detaillierte Diskussion ist wegen dem geschäftsspezifisch unterschiedlichen Bedarf wenig sinnvoll. Im Gegensatz dazu schafft die intensive Abhängigkeitsbetrachtung einen Rahmen für die Beantwortung von Anfragen im gesamten Infrastrukturmodell.

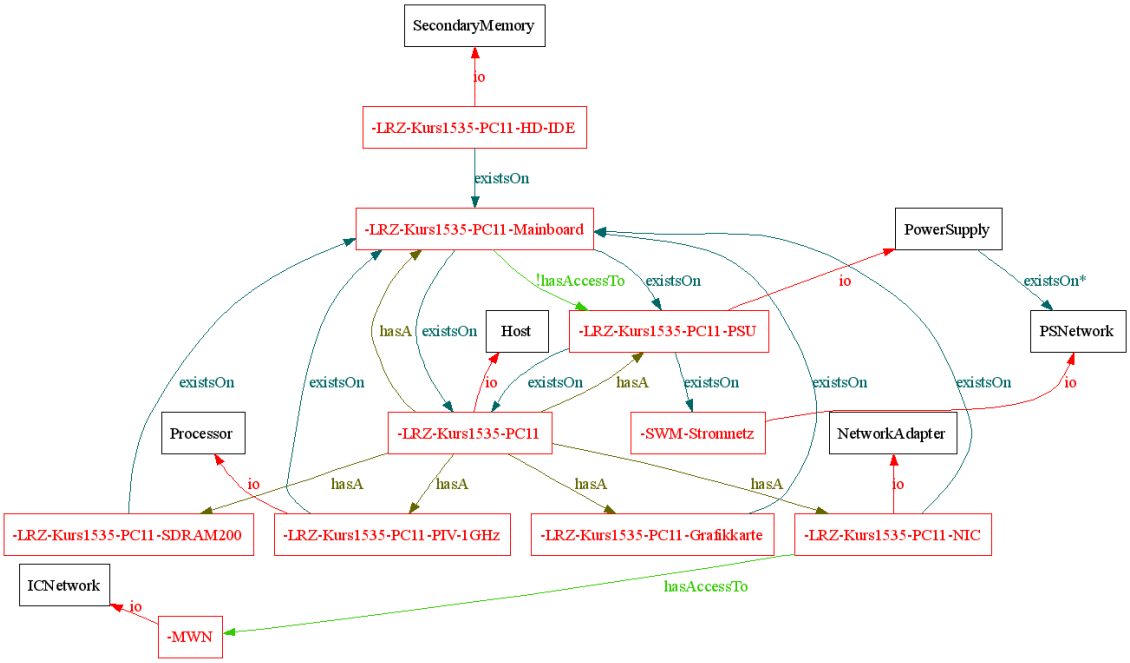


Abbildung 5-24: PC11-Komponenten

5.6.4.7 Beantwortung von Anfragen aus ITIL und MOF

Während auf den letzten Seiten ausführlich die Eigenschaften des Informationsmodells vorgestellt wurden, ist der ITIL- und MOF-Bezug etwas in den Hintergrund gerückt. Nun soll anhand von beispielhaften Anfragen aus ITSM Prozessen, deren Datenaustausch mit einer CMDB in Abbildung 5.2.1 dargestellt wurde, die zweckdienliche Verwendbarkeit des Informationsmodells vermittelt werden. Die Beispiele sind bewusst nicht allgemein gehalten, nachdem davon auszugehen ist, dass die Abfragen an eine CMDB individuell ausfallen, unabhängig vom anfragenden ITSM Prozess.

1. Abfrage der Festplatten eines gewissen Typs in allen im Betrieb befindlichen, kritischen Systemen, nachdem derartige Festplatten gehäuft Ausfallerscheinungen zeigen:
   Über das CI-Attribut *businessvalue* und allen davon kommunikationsabhängigen CIs, kann eine Prioritätenliste der CIs erstellt werden (=kritische Systeme), in der nach CIs der Klasse $SecondaryMemory$ gesucht wird. Falls $lifecycle\_state! = archived$, wird in der zugeordneten Klasse $AssetItem$ das Attribut $modelNumber$ sowie der *supplier* überprüft und gegebenenfalls eine gefährdete Festplatte gemeldet.

2. Nachdem ein grundlegender Change am Betriebssystem eines Rechners vorgenommen werden muss, müssen alle zugreifenden Benutzer sowie Administratoren über die Änderung und Downtime informiert werden:
   Auswahl der zugreifenden Benutzer über die Verfolgung aller Abhängigkeiten auf das Betriebssystem, auf die darauf installierten Programme, zugreifende Prozesse und wiederum zugreifende Benutzer im UserLayer. Auswahl aller Administratoren über die Verfolgung der *hasResponsible*-Abhängigkeiten der für die Benutzerauswahl ausfindig gemachten CIs.

3. Als Bestandteil einer Profitabilitätsprüfung von Service-Verträgen, sollen alle Service-bezogenen CIs mit Anschaffungskosten gezeigt werden, in die innerhalb eines festgelegten Zeitraums investiert wurde:
   Aus der Klasse *ServiceLevelAgreement* aus dem *UsageKnowledgeLayer* sind die einzubeziehenden SLAs auszuwählen und direkt sowie indirekt untergeordnete CIs zu bestimmen. Über die Historie in der CMDB sowie zugeordnete *AssetItems* lässt sich berechnen, welche Ausgaben für die CI-Beschaffung innerhalb des festgelegten Zeitraums für die Erhaltung der Services getätigt werden mussten.

Weitere Anfragen, für die eine CI-Selektion und Attribut-Aggregation im vorgeschlagenen Informationsmodell denkbar ist:

1. Welche Abschreibung hat die eingesetzte IT-Infrastruktur im aktuellen Jahr?
2. Wieviel kostet die IT-Infrastruktur für Abteilung x?
3. Für welche Komponenten verfällt in einem festgelegten Zeitraum die Garantie?
4. Wieviel Watt Leistungsaufnahme haben alle Rechnernetzteile in Raum x?
5. Welche Server haben überdurchschnittliche Auslastungsspitzen?
6. Welche Prozessbestandteile haben welche Availability?
7. Mit welcher Wahrscheinlichkeit fallen Komponenten aus bzw. wo sind Redundanzen sinnvoll?
8. Welche Komponenten erzeugen überdurchschnittlich viele Incidents/Problems und müssen daher optimiert werden?

Die möglichst einfache Verwaltung des Informationsmodells durch Komplexitätsreduzierung in Verbindung mit der möglichst weitgehenden Beantwortung von Anfragen zur IT-Infrastruktur, bestimmt letztendlich die Zweckdienlichkeit des Vorschlags. Insbesondere für die Bewertung der adäquaten Beantwortung von Anfragen ist eine generelle Aussage kaum möglich, da unklar ist, welche Informationen ITIL und MOF Prozesse im Detail benötigen (vgl. 5-2). Die Beantwortung der oben genannten Fragestellungen kann insofern nur einen Eindruck über die Mächtigkeit des Modells vermitteln. Nach eigener Auffassung wird mit dem Vorschlag den Anforderungen von ITIL und MOF in geeigneter Weise nachgekommen. Um diese Einschätzung zu fundieren, wird im Folgenden das Informationsmodell nach einem anerkannten Bewertungskatalog begutachtet.

5.6.5 Bewertung des Informationsmodells nach PinkVerify

Die Organisation Pink Elephant stellt als anerkanntes ITIL-Consulting-Unternehmen einen Fragenkatalog 'Mandatory, Integration and Functional Criteria' zur Bewertung von Tools zur Unterstützung des Configuration Managements unter [PE01] zur Verfügung. Ähnliche Bewertungssysteme wurden durch das itSMF unter [OGC03] für das Configuration Management nach ITIL und durch Microsoft unter [MOF03], unter anderem für die SMF 'Configuration Management', veröffentlicht. Beide sind allerdings nicht Tool-spezifisch gehalten, so dass an dieser Stelle zur Bewertung des Informationsmodells, mit hypothetischer Umsetzung in einem CMDB-Tool, die Kriterien aus PinkVerify heran gezogen werden. Fragestellungen, die sich auf die konkrete Umsetzung einer CMDB beziehen und nicht mit dem Informationsmodell in Verbindung stehen, wurden weg gelassen.

1. *Unterstützt das Tool die Registrierung und das Management von CIs (beispielsweise Hardware, Software, Contracts, SLAs) einer Organisation?*
   Zutreffend: CIs können je nach Bedarf erfasst werden, sind nicht beschränkt auf vorgegebene CI-Typen und können durch die Definitionen der Abhängigkeiten mit reduziertem Umfang in Verbindung gesetzt werden. Der Verwaltungsaufwand ist also im Vergleich zur Erfassung eines unstrukturierten Konfigurationsgraphen (vgl. 5-1) reduziert.

2. *Unterstützt das Tool die Aufnahme von CI-Attributen wie beispielsweise Seriennummer, Version oder Ortsangaben?*
   Zutreffend: Über die nach ITIL und MOF vorgeschlagenen, generellen Attribute eines $ConfigItems$ hinaus (vgl. Festlegung unter 5.6.2), können weitere Attribute in einem Vererbungsbaum des $ConfigItems$ frei ergänzt werden.

3. *Unterstützt das Tool die automatische Validierung der CI-Daten? Sind beispielsweise alle CI-Namen eindeutig?*
   Zutreffend: Neben der eindeutigen Prüfung der Namen wird auf notwendige und zulässige Attribute und Abhängigkeiten geprüft.

4. *Unterstützt das Tool die Bildung von Abhängigkeiten zwischen CIs? Beispielsweise parent/child, peer-to-peer, upstream/downstream.*
   Zutreffend: Abhängigkeiten sind fest definiert und zulässige Klassen von Zielobjekten CI-spezifisch festgelegt.

5. *Unterstützt das Tool ein anpassbares Statusmanagement für den CI Lebenszyklus? Beispielsweise geplant, bestellt, in Entwicklung, im Test, implementiert, in Produktion, im Wartungszustand.*
   Zutreffend: die vorgeschlagenen Zustände eines CI Lebenszyklus unter 5-16 sind nicht festgelegt.

6. *Unterstützt das Tool ausschließlich autorisierten Zugriff auf die CMDB für Lese- und Schreibzugriffe?*
   Zutreffend: Die Aufnahme von Verantwortlichkeiten zu einzelnen CIs ermöglicht eine dedizierte Zugriffssteuerung.

7. *Unterstützt das Tool die Aufnahme von CI Baselines? Beispielsweise für die Rücksetzung auf eine vorhergehende Version für den Fall, dass ein Change misslingt.*
   Zutreffend: die Aufnahme der CI-History für alle CIs ermöglicht die Generierung einer CI baseline zu jedem gewünschten Zeitraum über die Dauer der Historie und erlaubt eine entsprechende Rücksetzung.

8. *Unterstützt das Tool die Protokollierung von Änderungen eines CIs für Audits? Beispielsweise das Installationsdatum, Änderungsdaten und vorhergehende Unterbringung.*
   Zutreffend: vergleiche vorhergehende Frage.

9. *Stellt das Tool flexible Auswertungen bereit, die sich auf den CI Bestand, Anlagegüter und Finanzinformationen beziehen, um Audits zu unterstützen?*
   Zutreffend: Die vielfältigen Abfragemöglichkeiten des Informationsmodells bieten eine geeignete Basis zur Entwicklung derartiger Reports.

10. *Unterstützt das Tool das Incident Management mit der Bereitstellung von kritischen und defekten CIs zur Klassifizierung der Incident Records?*
    Zutreffend: Die Funktionstüchtigkeit kritischer und defekter CIs wird nach Meldung eines Fehlers für alle abhängigen CIs gesetzt. Nachdem Incidents CIs zugeordnet werden können, kann die Störungsinformation zur Klassifizierung verwendet werden.

11. *Unterstützt die CMDB proaktives Problemmanagement durch die Identifizierung von problematischen oder instabilen Komponenten? Stellt beispielsweise das CI Status Accounting Informationen über störanfällige und wartungsintensive CIs bereit?*
    Zutreffend: Die vorgeschlagenen CI-Attribute ermöglichen die Erfassung von instabilen Komponenten ($probability\_non\_working$). Die Auswertung der Anzahl auflaufender Incidents für CIs oder der Frequenz der Status-Änderungen trägt zur Identifizierung störanfälliger CIs bei.

12. *Unterstützt das Tool die Begutachtung und Genehmigung von Change Requests über die Bereitstellung von Informationen betroffener CIs?*
Zutreffend: Über die beschriebene Impact-Analyse werden für den Review-Prozess entsprechende CIs zur Überprüfung vorgeschlagen.

13. *Unterstützt das Tool die Identifizierung verschiedener RFCs die sich auf das gleiche CI beziehen?*
Zutreffend: Sofern einzelnen RFCs zu CIs zugeordnet werden.

14. *Unterstützt das Tool die Aufnahme von CI-Status-Änderungen, sobald Änderungen vorgeschlagen oder autorisierte Änderungen implementiert werden? Beispielsweise den Grund der Änderung, den zukünftigen Status, den geplanten Statuswechsel.*
Zutreffend: Statusänderungen mit zugehörigen Beschreibungen können für einen beliebigen, zukünftigen Zeitpunkt eingetragen werden, sofern autorisiert.

15. *Können innerhalb des Tools Anpassungen der CI Abhängigkeiten vorgenommen werden, um organisationsspezifischen Anforderungen nach zu kommen?*
Nicht zutreffend: dieser Punkt soll bewusst vor dem Hintergrund der Designentscheidung für die Abhängigkeitsreduktion unterbunden bleiben.

16. *Unterstützt das Tool die automatische Wiederherstellung von Parent- und Child-Beziehungen sofern CIs erfasst, geändert oder gelöscht werden?*
Zutreffend: Die festgelegten Abhängigkeiten lassen insbesondere beim Auflösen von CIs das automatische Setzen der Abhängigkeiten umgebender CIs zu. Bei Aufnahme neuer CIs ergeben sich individuell zu setzende Abhängigkeiten, die beispielsweise beim Einsatz von CI-Templates automatisch gesetzt werden können.

5.6.6 Zusammenfassende Modellbetrachtung

Das vorgeschlagene Informationsmodell bietet die flexible Erfassung identifizierter Infrastrukturelemente mit räumlicher und zeitlicher Einordnung (beispielsweise über die Modellierung von Prozessen oder über die CI-Historie), Zuordnung von Nutzungsrechten und Verantwortlichkeiten, Dokumentationen, Finanzinformationen sowie Gruppierung beziehungsweise Klassifizierung der CIs. Diese Dimensionen der Datenerfassung ermöglichen die umfassende Modellierung der IT-Infrastruktur sowie vielfältige Abfragemöglichkeiten aus ITSM Prozessen. Die Modellflexibilität bezieht sich zum einen auf die Möglichkeit, die Erfassung auf einzelne Ebenen des Layermodells oder einzelne Servicegraphen zu beschränken. Zum anderen kann die Detaillierung der Objektmodellierung sowie der Attributierung vergleichbarer Komponenten variieren. Damit kann das Modell je nach Bedarf ausgestaltet werden. Die Anforderungen von ITIL und MOF, die unter 3.3.1 zusammengefasst wurden, werden nach 5.6.5 erfüllt.
Die Reduzierung der Abhängigkeiten schafft in Verbindung mit der Ebenenbildung eine Reduktion der Modellkomplexität im Vergleich zur Vielfalt der Erfassungsmöglichkeiten im unstrukturierten Konfigurationsgraphen (5-1). Die Einschränkung ist allerdings genau zu prüfen, beispielsweise ergab sich aus der Szenario-Modellierung, dass insbesondere mit den festgelegten Abhängigkeiten existsOn und dependsOn in Bezug auf nicht-kommunizierende Komponenten oder bei der Workflow-Modellierung der Zugriff aufwändig beschränkt werden muss, um eine gültige Impact-Analyse betreiben zu können. Letztendlich war trotz der Beschränkung der Szenarioerfassung und der Designentscheidung für eine weitgehende Komplexitätsreduzierung der Modellierungsaufwand relativ hoch. Es scheint, als ob eine Beschränkung auf diejenigen Komponenten sinnvoll ist, auf die im Rahmen administrativer Tätigkeiten oder bei der Benutzung im ersten Schritt zugegriffen wird. Beispielsweise sind an einem Rechensystem nach außen hin die Hardware, das Betriebssystem und die vorrangig verwendeten Softwarepakete von Interesse, nicht aber einzelne Softwaretools, laufende Prozesse und einzelne Dateien. Werden von letzteren Informationen benötigt, wird die Infrastrukturkomponente direkt untersucht, statt eine CMDB abzufragen. Eine klare Abgrenzung dieser Fälle ist allerdings wegen der Vielfalt an Komponenten sowie der variierenden Nutzungsintention nicht möglich.

Im Gespräch mit Administratoren und Beratern wurde deutlich, dass der Nutzwert einer CMDB ausdrücklich von der Möglichkeit der automatischen Datenerfassung (gefiltert) abhängt, nachdem dadurch bei etwa gleichbleibendem Aufwand die Vielfalt und Detailgenauigkeit der erfassten Daten bei häufigeren Aktualisierungsintervallen entscheidend gesteigert werden kann - für ein besseres Kosten-/Nutzen-Verhältnis. Die Organisation des Aufbaus einer CMDB sowie der Pflegeprozesse wird im nächsten Kapitel beschrieben.

6 Arbeitsprozessmodell für Configuration Management

Nachdem nun ein verhältnismäßig konkreter Vorschlag für ein Informationsmodell zur Ableitung einer CMDB vorgestellt wurde, kann nun auf dieser Basis die Organisation der Leistungen zur Bereitstellung und Verwaltung der Konfigurationen in einer CMDB präsentiert werden. Dies geschieht anhand eines erarbeiteten Referenzprozessmodells, das folgende Funktionen erfüllen soll. Zum einen sollen über praxisnahe Beschreibungen für das Training von Configuration Managern und anderer CM-Beteiligter zügig erfassbare und praxisrelevante Arbeitsbeschreibungen bereit gestellt werden. Zum anderen sollen die generalisierten Prozessbeschreibungen zur Ableitung von konkreten, unternehmensspezifischen Prozessen dienen. Das Arbeitsprozessmodell wurde aus der Kombination der Vorgaben aus der ITIL, dem MOF (Microsoft Operations Framework), Unterlagen eines Microsoft-RZ in Dublin, einem Arbeitsprozessmodell 'Configuration Manager' des Fraunhofer Instituts ISST [Saa03], den Ergebnissen aus der Befragung des LRZ zum Configuration Management [LRZ04] und eigenen Erfahrungswerten zusammengestellt. Zunächst werden im nächsten Kapitel Grundlagen für die Modellierung betrachtet, mit denen die in Folge vorgestellten Referenzprozesse ausgearbeitet wurden.

6.1 Modellierungsgrundlagen

6.1.1 Das ARIS-Modell

Die Verwendung des ARIS-Konzepts (Architektur integrierter Informationssysteme) von Prof. August-Wilhelm Scheer (Institut für Wirtschaftsinformatik an der Universität des Saarlandes) bei der Planung betrieblicher Informationssysteme vermittelt eine Gesamtsicht der notwendigen Systembestandteile und deren Interaktion vor dem Ziel der vollständigen Berücksichtigung von Anforderungen. Das ARIS-Konzept zur Unterstützung der Prozessmodellierung basiert hauptsächlich auf einer Sichten-Architektur, dem ARIS-Haus 6-1.

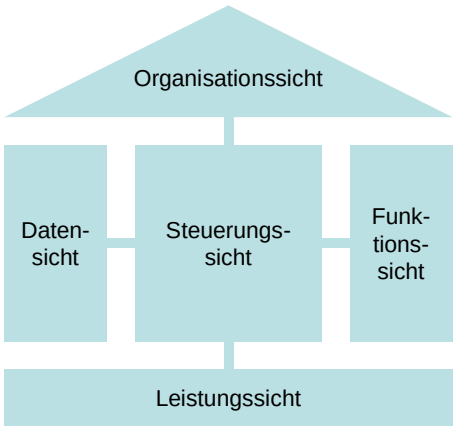


Abbildung 6-1: ARIS-Haus

Die Funktionssicht beschreibt alle funktionalen Elemente, die für die Verrichtung an Objekten zur Unterstützung von Zielen vorhanden sind, während die Organisationssicht alle Organisati-

onseinheiten wie menschliche Arbeitskräfte, Maschinen und Hardware beschreibt. Die Datensicht enthält alle Ereignisse, die Daten mit entsprechenden Umfelddaten generieren. Die Leistungssicht betrachtet alle ein- und ausgehenden Dienst-, Sach- und finanziellen Leistungen. In der Steuerungssicht werden die vorangegangenen Sichten in einem logischen und zeitlichen Ablaufplan integriert [GNU05, AS05].

Um die Tätigkeiten zum Aufbau und zur Pflege des dargestellten Informationsmodells zu vermitteln, wird in den folgenden Kapiteln als Referenz ein Arbeitsprozess eines Configuration Managers mit Gesamtverantwortung für das Configuration Management, auf der Beschreibungsebene des IT Service Managements, vorgestellt. Die einzelnen Bestandteile sind geschäftsspezifisch möglicherweise mehreren Verantwortlichen zu übertragen (nach ITIL: Management Staff) oder mehrfach für verschiedene Bereiche der IT-Infrastruktur aufzubauen. Im Arbeitsprozessmodell sind verschiedene Sichten zusammengelegt. Bei Zerlegung gemäß dem ARIS-Konzept kann Folgendes vorausgreifend festgestellt werden (ähnlich zur Bewertung nach dem OSI-Modell unter 3.3):
Die Organisationssicht fällt wegen der Kombination der oben genannten Rollen innerhalb des Configuration Managements vereinfacht aus. Rollen aus anderen ITSM Prozessen, die mit dem Configuration Management interagieren, sind in einer flachen Hierarchie mit Leitung durch die 'IT-Strategie' organisiert. Die sonstigen Ressourcen sind in keiner nennenswerte Form organisiert. In der Datensicht sind alle Ereignisse enthalten, wie beispielsweise die Anmeldung von Bedarf an gewissen Daten des Configuration Managements sowie Ergebnisse, beispielsweise eine, zu einem gewissen Zeitpunkt fertig gestellte CMDB. Bestandteile, die der Leistungssicht zuzuordnen sind, befinden sich kaum im Arbeitsprozessmodell während in der Funktionssicht alle Arbeitsschritte des Modells enthalten sind. Die Gesamtsicht aller relevanten Bestandteile gemäß dem ARIS-Konzept wurde in dem kombinierten Arbeitsprozessmodell folglich berücksichtigt.

6.1.2 Auswahl der Modellsprache

Gemäß [BSIG00] sind nach den GoB die Sichten sprachneutral, das heißt, für eine Prozessmodellierung können unterschiedliche Sprachen wie EPK, Zustandsübergangsdiagramme, Petri-Netze und andere verwendet werden. Für die ARIS-Modellierung werden im gewöhnlichen EPK-Diagramme verwendet. In dieser Arbeiten wurden jedoch zur Visualisierung UML-Activity-Diagramme ausgewählt, nachdem damit die bei EPK-Diagrammen häufige Wiederholung von aktionsfolgenden Ergebnissen aus ausreichend beschreibenden Aktionen vermieden wird. Gewöhnlich werden die Aktiviäten in Activity-Diagrams mit Verben beschrieben. Aus Gründen der oftmals kürzeren Schreibweise wurden in den folgenden Diagrammen stattdessen Substantive verwendet, was die Verständlichkeit allerdings nicht beeinträchtigen sollte.

6.2 Der Referenzprozess im Überblick

Die in Abbildung 6-2 dargestellten Bahnen (UML-Terminologie: swimlanes) enthalten am oberen Ende die Nennung der verantwortlichen ITSM-Prozesse für die zugeordneten Aktivitäten. Die IT-Strategie delegiert in der ersten Initiative die Planung von CM an den Prozessverantwortlichen für das Configuration Management. Später nimmt die IT-Strategie Auswertungen über den Erfolg des Configuration Managements entgegen und weist gegebenenfalls Optimierungen des CM an, die letztendlich einer Überarbeitung der vorangegangenen Planungsaktivitäten entsprechen. Deshalb wurden innerhalb eines Arbeitsprozesses die **Planung und Optimierung von CM** (6.3.1) zusammengefasst. Dieser Arbeitsprozess erstreckt sich bis zur Einführung und **Freigabe einer CMDB**.
Nun können die Informationen in der CMDB aus den ITSM-Prozessen abgefragt werden. Wie bereits früher festgestellt wurde, gibt es keine nennenswerten, prozesspezifischen CMDB-Abfragen. Deshalb wurden die ITSM-Prozesse für den Arbeitsprozess der **CMDB-Abfragen** (6.3.3) in einer Bahn zusammengefasst. Das Release- und Change-Management interagieren mit dem Configura-

tion Management für die Einbringung von Änderungen, die im Arbeitsprozess **CMDB-Changes** (6.3.2) beschrieben werden. In diesem Arbeitsprozess erfolgt die differenzierte Betrachtung nach beiden Prozessen. Des Weiteren wird in den detaillierten Arbeitsprozessmodellen das Incident-Management für die Entgegennahme von Incidents aus dem Configuration Management separat mit aufgenommen. Sowohl CMDB-Abfragen als auch CMDB-Changes werden durch den Arbeitsprozess **CM-Review** (6.3.4) überwacht. Dabei wird für die operativen Tätigkeiten des CM vorrangig eine Auswertung nach KPIs betrieben (5), um gegebenenfalls eine Optimierung des Configuration Managements einzuleiten.

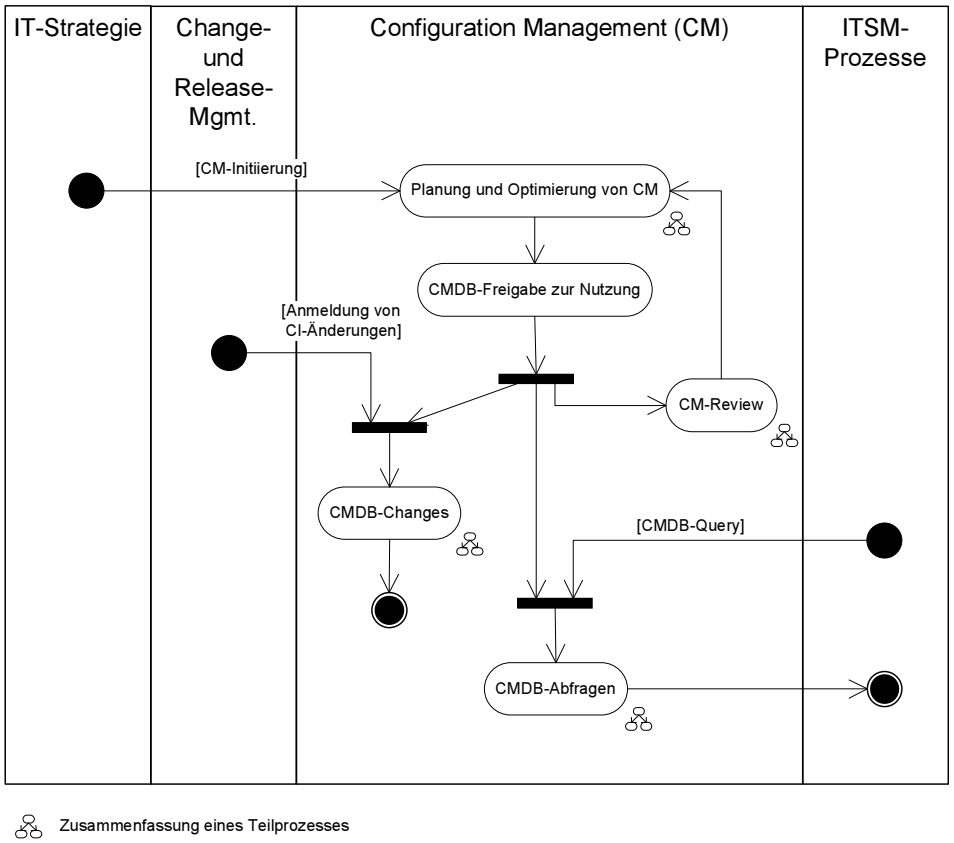


Abbildung 6-2: Der CM-Arbeitsprozess im Überblick

6.3 Teilprozesse des Arbeitsprozessmodells

Die oben dargestellten Teilprozesse werden nun in der Reihenfolge Planung und Optimierung des CM, CMDB-Changes, CMDB-Abfragen und CMDB-Review beschrieben. Nachdem die kritischen Faktoren bei der Umsetzung von Configuration Management (3.4) insbesondere den Planungs- und Optimierungsprozess betreffen, wird im Folgenden auf diesen Bestandteil Hauptaugenmerk gelegt, während die restlichen operativen Tätigkeiten in Ergänzung knapp abgehandelt werden können. Jeder Arbeitsschritt unterliegt dem Monitor-Control-Loop (2-5), das heißt, dass die Projektbestandteile ständig optimiert werden (1(a)i). Die Optimierung umfasst die Entscheidung über einen Abbruch, sofern keine Rentabilität prognostiziert und bestätigt werden kann.

6.3.1 Planung des Configuration Managements

Wie schon eingangs erwähnt, ist das Problem des Managements, möglichst (kosten-)effizient Managementdaten bereit zu stellen, um gleichzeitig möglichst effektiv managen zu können. Im letzten Kapitel wurde deutlich, dass dieses Optimierungsproblem bedarfsbedingt und damit individuell zu lösen ist (5.6.6). Der in Veränderung befindliche Bedarf an Managementinformation sowie sich verbessernde Methoden zur Bereitstellung von Daten, verändern laufend das Optimum.

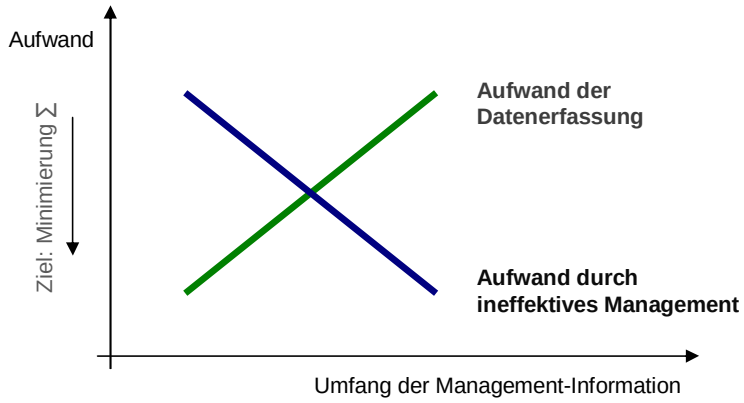


Abbildung 6-3: Optimierungsproblem des Managements

Zur Annäherung an das Optimum müssen also die Aufwandsgrössen bekannt sein. Ein IT-eigenes Controlling schafft die Möglichkeit, Kosten und Leistungen zu überwachen. Der **Datenerfassungsaufwand** kann über die Protokollierung der Arbeitszeiten für die Durchführung sowie die Kosten für Investitionen zur Unterstützung des Configuration Managements berechnet werden. Der Aufwand, der notwendig ist, **ineffektives Management** auszugleichen, ist mangels Kenntnis des effektiven Managements selbst nicht bestimmbar. Abbildung 6-3 wurde insofern nur für die Darstellung des grundsätzlichen Zusammenhangs aufgenommen. Ein anwendbares Maß stellt allerdings die Differenz der Kosten in den iterativen Optimierungszyklen dar - also beispielsweise der Arbeitsaufwand für gewisse ITSM Aktivitäten vor und nach der Einführung des Configuration Managements. Im Folgenden wird davon ausgegangen, dass es ein IT-Controlling gibt, in dem die Zuordnung von Aufwand zu einzelnen Komponenten der IT-Infrastruktur betrieben wird (wer, wann, warum, was wo und wie, vgl. 5.4.2), um den Optimierungsvorgang zu unterstützen.

6.3.1.1 Initialisierung

Der unter 6-4 dargestellte Planungsprozess kann erfahrungsgemäß nicht strikt durchlaufen werden. Stattdessen empfiehlt es sich, nach Bedarf Rücksprünge durchzuführen. **Businesstreiber** bestimmen sich durch die Vorteile des Configuration Managements (3.1). Der **Bedarf an Konfigurationsinformationen** seitens der **ITSM Prozesse** wurde unter 5-2 heraus gearbeitet. Nach

Entscheidung für die Planung des Configuration Managements seitens der **IT Strategie**, werden durch die Projektleitung **Problemstellungen** und Ziele gesammelt und in einer **Vision zum CM** formuliert. Die Projektvision kann an Kennzahlen des IT-Controllings festgemacht werden. So ergeben die Beträge unter verschiedenen Kostenstellen und Kostenarten Ansatzpunkte für die Abschätzung von Optimierungen.

6.3.1.2 Analyse

Zur engeren Kooperation und Kommunikation werden folglich Ansprechpartner aus **betroffenen Personengruppen analysiert**, ausgewählt und an den **Planungen beteiligt**. Es bietet sich zur Erzielung von 'Quick Wins' an, mit denjenigen Personengruppen zu beginnen, die den größten, tatsächlichen Bedarf an Konfigurationsinformation für einen überschaubaren Bereich angemeldet haben (1(a)iii). Bei der Entwicklung des Configuration Managements sind diejenigen Anwender am erfolgreichsten, die innerhalb einzelner Optimierungszyklen verhältnismäßig geringe Erweiterungen einbringen [Jak04]. Gleichzeitig können **Fachinformationen einbezogen** werden, um den Planungsprozess abzukürzen und zu optimieren. Sowohl mit den ausgewählten, internen Ansprechpartnern (1(f)ii) als auch mit gegebenenfalls hinzu gezogenen Beratern, können die Ziele und die Vision des CM konkretisiert und festgelegt werden. Die Abschätzungen zu Kostenoptimierungen einzelner Kostenstellen und Kostenarten können durch die Analyse vergleichbarer Projekte, beispielsweise nach ROI, abgesichert werden.
Die **Festlegung der Grenzen des CM** entspricht zum einen den Überlegungen zum Informationsmodell unter 5.2. Damit werden die Zuständigkeiten für die Datenhaltung unter ITSM Prozessen festgelegt. Zum anderen werden innerhalb der Infrastruktur Teilbereiche ausgewählt, deren Konfigurationen zunächst aufgenommen werden sollen.
Vorhandene CMDBs sind beispielsweise eine Mitarbeiterliste aus dem Sekretariat, eine Aufstellung der Anlagegüter aus der Finanzbuchhaltung sowie durch einzelne Administratoren geführte Tabellen mit Rechner-Ansprechpartner-Relationen. Aus diesem gesammelten Datenmaterial sind Konfigurationen ersichtlich, die offensichtlich einem vorhandenen Bedarf entsprechen. Sie dienen als Grundlage für eine spätere, erste Befüllung einer zusammenfassenden CMDB, welche die ITIL-Zielobjekte 'People, Process und Technology' in Verbindung setzt.
In der **Infrastrukturanalyse** erfolgt die Überprüfung vorhandener Komponenten vor dem Hintergrund der Kenntnis verschiedener Informationsmodelle (vgl. Kapitel 5). Die **Analyse und Priorisierung von Konfigurationsanfragen** nimmt eine zentrale Position bei der bedarfsgerechten Entwicklung eines Informationsmodells ein (1(a)iii). Die Sichtung historischer Informationen aus Incident- und Problemtickets in einem Help-Desk-System ist hilfreich für die Auswahl von häufig benötigten Konfigurationsinformationen.

6.3.1.3 Spezifikation

Mit dem Vorwissen aus der Analysephase kann nun mit der **Entwicklung von Standards** für das CM begonnen werden. Dies betrifft einerseits organisatorische und prozessbezogene Standards. So sollten **Pflegeprozesse** festgelegt werden, für die in den Abbildungen 6-6, 6-7 und 6-8 geeignete Referenzprozesse zur Verfügung stehen. Des Weiteren sollten **Rollenbeschreibungen** verfasst werden, welche die Verantwortlichkeiten und Rechte für zuzuordnende Prozess- und Infrastrukturbestandteile regeln. ITIL und MOF schlagen die Rollendefinition eines vollberechtigten Configuration Managers und eines zuarbeitenden Configuration Management Staffs vor. Andererseits sollten Standards für die Infrastrukturbeschreibung über CIs eingeführt werden. Das im Kapitel 5 vorgeschlagene Informationsmodell ist dafür eine mögliche, ITIL- und MOF-konforme Lösung. Die **Komponentenattributierung** beziehungsweise CI-Attributierung ergibt sich aus entsprechenden Anfragehäufigkeiten aus der Anfragenanalyse. Wie im vorhergegangenen Kapitel erläutert wurde, bietet es sich unter anderem aus Gründen des besseren Imports von Daten aus Discovery-Tools an, ein einfaches Mapping der Attribute mit einem Modellierungsstandard wie CIM aufzunehmen. Die **Strukturierung** der CIs sowie die Bestimmung von **Abhängigkeiten** wurde unter 5.5 ausführlich diskutiert. **Namenskonventionen** sollten Bezugsobjekte und gege-

benenfalls Versionsinformationen enthalten, um ein CI über den Namen in einen strukturellen und zeitlichen Zusammenhang setzen zu können. Neben der Bedarfsorientierung über die Anfragenanalyse lassen sich bei der Festlegung von **Identifikationsregeln** die folgenden Maßstäbe ansetzen:

- Die Betrachtung von Servicegraphen, vgl. Abbildung 2-2, ermöglichen die Bestimmung aller IT Infrastrukturkomponenten mit zuständigen Administratoren, Benutzern, zugeordneten Prozessen usw., welche die Bereitstellung des Services ermöglichen. Die Einordnung in einen Servicegraphen kann also als Identifikationsregel für ein zu erfassendes CI verwendet werden. In der Modellierung des Szenarios aus Kapitel 4 hat sich ergeben, dass selbst bei Betrachtung einzelner IT Services nach dieser Regel im Allgemeinen weite Teile der Basis-Infrastruktur einbezogen werden müssen.

- Die Beschränkung auf die Verantwortlichkeit ausgewählter Personen kann als weitere Identifikationsregel angewandt werden. Eine Abgrenzung findet dabei über die alleinige Verantwortung für die auszuwählenden Komponenten statt, das heißt, es gibt keine Verträge mit anderen Dienstleistern, die im Haftungsfall für die jeweiligen Komponenten heran gezogen werden können.

- Nach Aussage von [Loo04] wird bei der Auswahl von aufzunehmenden Komponenten meist ein minimalistischer Weg beschritten. Notwendige, wertvolle und geschäftskritische Komponenten werden erfasst, indirekt ersichtliche und gewöhnlich nicht direkt nutzbare Komponenten werden ausgeklammert. Beispielsweise werden die Steckkarten eines Rechners vernachlässigt, für den Rechner selbst aber ein CI angelegt. Diese Aussage bestätigt die primäre Aufgabe des Configuration Managements in der Praxis, den übergeordneten Zusammenhang der Infrastrukturbestandteile bereit zu stellen. In Umsetzungsprojekten ist die Identifikation von geeigneten CIs im Allgemeinen keine größere Herausforderung, nachdem der Bedarf und der Nutzen bereitgestellter CI-Informationen für Anwender meist offenbar ist [EXP].

Nun können die **Ziele und Standards des Configuration Managements dokumentiert** (siehe 6-5) und weitere Spezifikationen ausgearbeitet werden. In ITIL und MOF wird vorgeschlagen, **KPIs festzulegen** (1g), die insbesondere Fehler überwachen, welche sich in Verbindung mit den 'kritischen Faktoren' (3.4) einstellen. Um das Gesamtergebnis einer CM-Prozesstransformation zu prüfen, kann wie erwähnt auf das IT-Controlling zurück gegriffen werden. Entsprechende KPIs sind also auch auf Kostenträger und Kostenstellen anzusetzen. In das **CMDB-Design**, also die Spezifikation für die Umsetzung oder Auswahl einer CMDB-Softwarelösung, gehen die oben genannten Standards zum CMDB-Informationsmodell ein, vgl. Kapitel 5. Durch das Projektmanagement können die definierten **Rollen mit Personen besetzt** (1(f)i) und **Milestones** festgelegt werden. Zur Zusammenfassung der Spezifikationsphase wird ein **Configuration Management Plan** verfasst (1h), für den in der folgenden Tabelle beispielhaft eine Unterteilung gezeigt wird.

Configuration Management Plan

1. Einführung
   a) Vision
   b) Umfang
   c) Definitionen
   d) Bezugsquellen
   e) Überblick
2. Key Performance Indikatoren
3. Configuration Management Prozesse
   a) Planung

   b) CMDB-Changes
   c) CMDB-Abfragen
   d) Review
4. Rollenbeschreibungen und Besetzung
5. CMDB-Design
6. Schulungsinhalte
7. Milestones

6.3.1.4 Implementierung und Customizing

Bei der **Entwicklung einer CMDB** beziehungsweise bei der Auswahl einer CMDB-Softwarelösung, sollten über die genannten, CM-spezifischen Anforderungen hinaus, die folgenden Anforderungen berücksichtigt werden:

- Verfügbarkeit von Mediatoren zur Anbindung externer Datenquellen (Discovery-Tools, bestehende CMDBs)
- Bereitstellung einer mächtigen Abfragesprache
- Flexible Datenstrukturen für ständige Anpassungen des Datenmodells
- Unterstützung von Mass-Update-Funktionalität zur Vereinfachung von Änderungen
- Protokollierung der CMDB-Nutzung (beispielsweise des Zugriffs auf einzelne Items in der Datenbank) zum Zwecke der Auswertung und Optimierung

Um bei potentiellen Benutzern einer CMDB Verständnis für die Möglichkeiten des CMDB Einsatzes zu schaffen, sollte deren **Nutzen kommuniziert** werden, beispielsweise über **Personalschulungen** (1(f)ii). Während die CMDB-Nutzung insbesondere bei der Zielgruppe der IT Administratoren für diejenigen Konfigurationen, welche sich im Detail direkt an der IT Infrastruktur abfragen lassen, auf geringeres Interesse stößt, wird die Bereitstellung der übergeordneten Zusammenhänge von 'People, Process und Technology' als hilfreiche Information angenommen [EXP]. Für eine schrittweise 'Evolution des Configuration Managements' innerhalb eines Unternehmens bietet es sich also an, zunächst mit der Bereitstellung dieser übergeordneten Zusammenhänge zu beginnen.
Die **Identifikation von CIs** (2) wird bestimmt durch die spezifizierten Identifikationsregeln, das **Labeling von CIs** wird unter Berücksichtigung der festgelegten Namenskonventionen (2b) durchgeführt. Es bietet sich für eine schnellere Erfassung (2c) bei der Systeminventur sowie für die bessere, menschliche Lesbarkeit an, beim Labeling ein kombiniertes Verfahren aus Beschriftung und maschinell lesbaren Identifikatoren (beispielsweise Barcodes) zu verwenden. Die identifizierten Komponenten können nun entweder als CIs **manuell erfasst** oder aus **bestehende CMDBs** oder **Discovery Tools** importiert werden. Nach der Absicherung der CMDB (3d) kann die **CMDB zur Nutzung freigegeben** werden.

6.3.1.5 Auto-Discovery

Nachdem Discovery Tools wegen der Beschleunigung der Datenerfassung für die Ausweitung des CI-Umfangs, häufigere CI-Reviews und einer folglich höheren Datengüte eine wichtige Rolle zukommt (3c, 1(e)iv), werden in diesem Kapitel Beispiele für die automatischen Erfassung von CI-Objekten aus verschiedenen Ebenen des Informationsmodells aus Kapitel 5 beschrieben.

Für die Ebene der aktiven Netzwerkkomponenten hat sich mit SNMP ein Standard durchgesetzt, der relativ weitgehend den Austausch von Konfigurationsdaten von Netzwerkkomponenten zulässt.

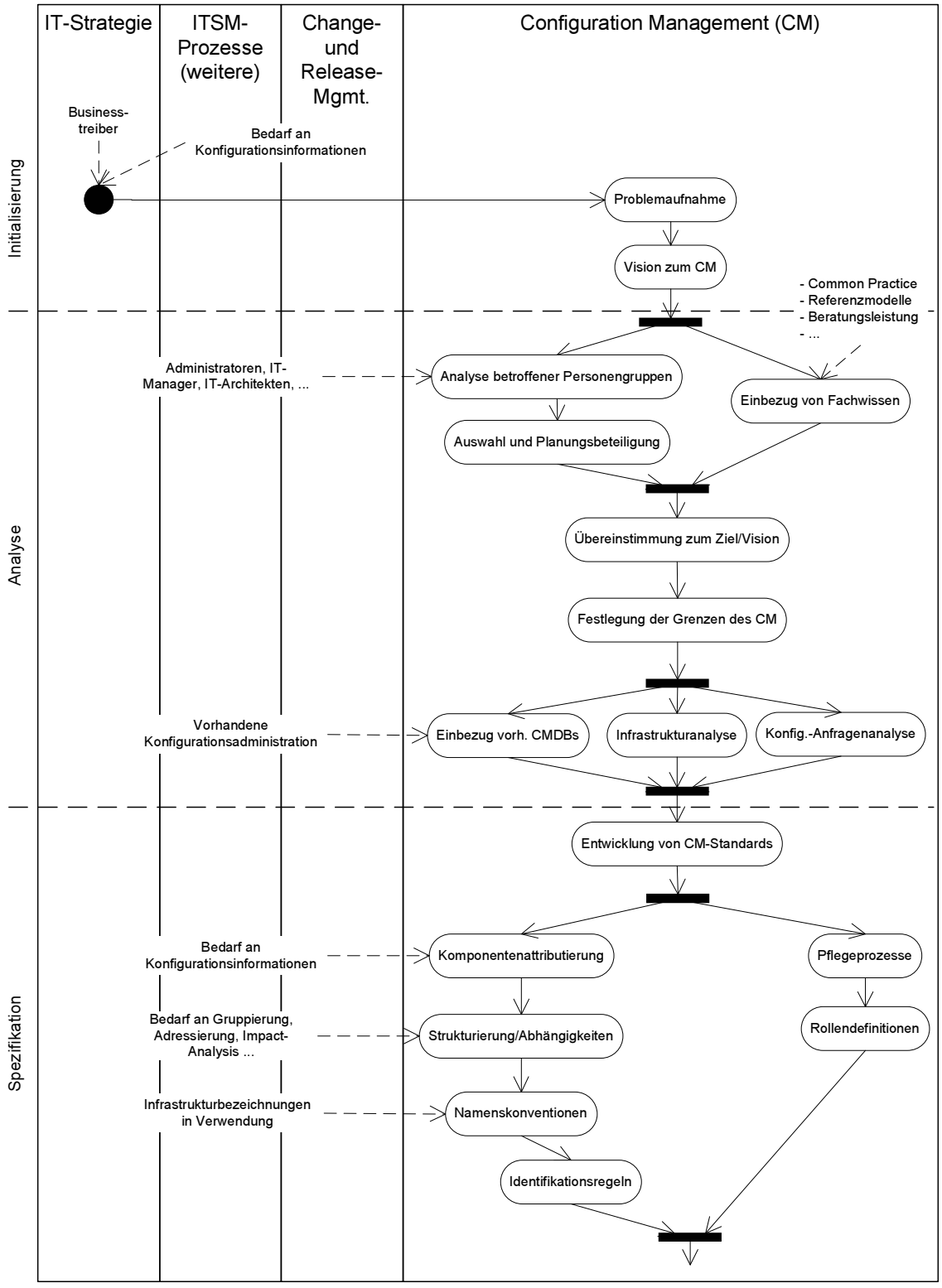


Abbildung 6-4: Arbeitsprozess CM-Planung (1)

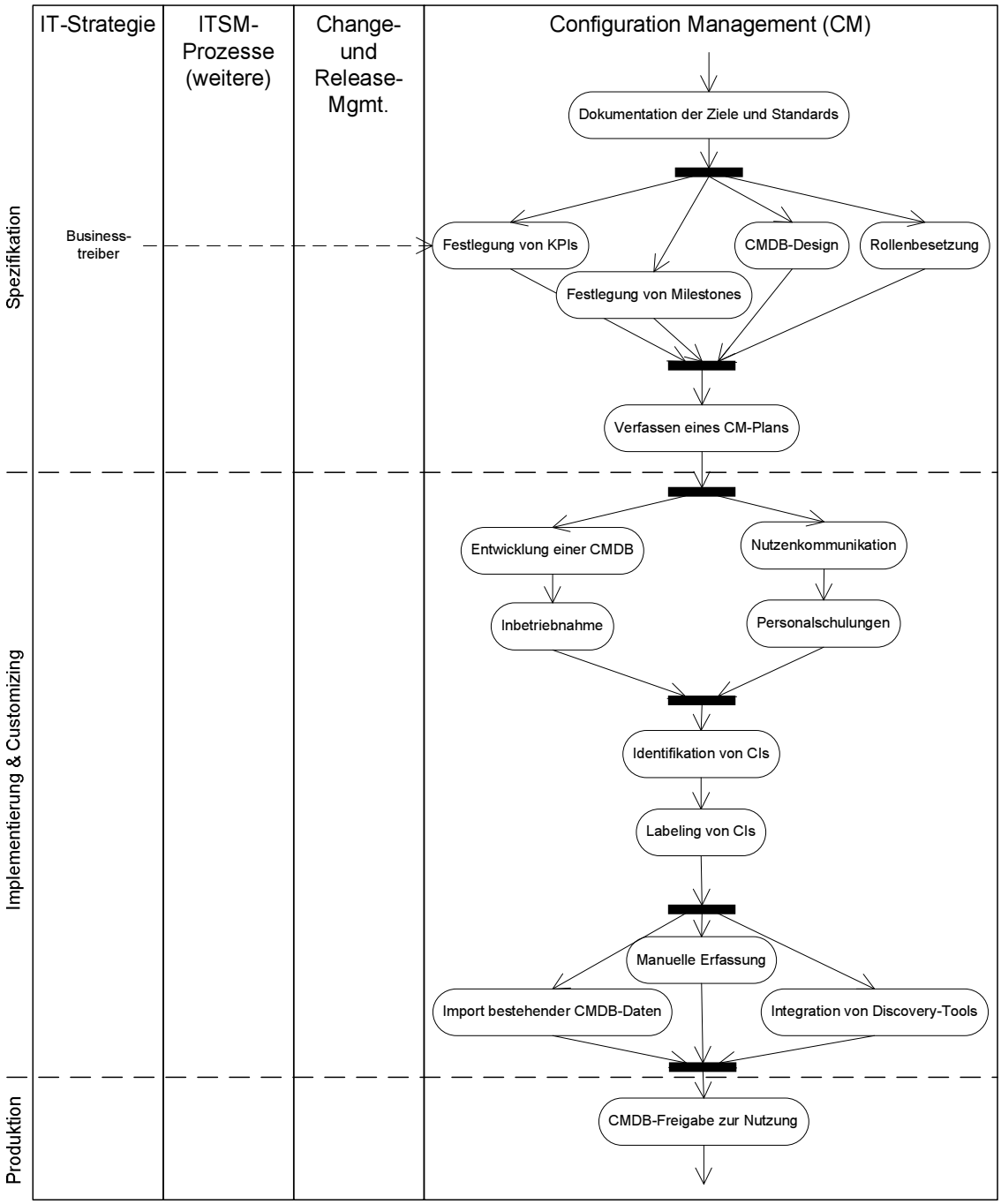


Abbildung 6-5: Arbeitsprozess CM-Planung (2)

In Kapitel 5.3.1 wurde der Einsatz von WBEM für den Austausch von Konfigurationsdaten aus dem CIM-Modell besprochen, welches insbesondere für die Verwaltung von Betriebssystemkonfigurationen verwendet wird (vgl. WMI für Windows-Systeme). Beide Standards werden intensiv dafür eingesetzt, um über Auto-Discovery-Tools Basisdaten für die Infrastrukturerfassung bereit zu stellen. Allerdings sind auch hier Verbesserungen notwendig. So wurde im Rahmen der Veranstaltung [Jak04] beispielsweise der Bedarf genannt, die physische Verbindung von Peripheriegeräten wie Drucker oder Scanner an einzelnen Arbeitsplätzen zentral und automatisch überprüfen zu müssen. Standards wie WMI lassen hierbei zu, einen vorhandenen Gerätetreiber auf dem Arbeitsplatzrechner abzufragen, nicht aber den tatsächlichen Anschluß und die Betriebsbereitschaft des Peripheriegeräts.
Betrachtet man über die Betriebssystemebene hinaus einzelne Softwareanwendungen, so ist kein durchgesetzter Standard für deren Management vorhanden. Speziell für Java-Anwendungen schaffen die Java Management Extensions (JMX) die Möglichkeit des einheitlichen Managements, beispielsweise bezüglich des Loggings von Events, der Messung der Laufzeit von Prozeduraufführungen und des Speicherbedarfs einzelner Prozesse. Diese Informationen werden bei einem weitergehenden Configuration Management benötigt, um insbesondere den Anforderungen der Impact-Analyse nachkommen zu können. Für ein besseres Anwendungsmanagement ist es folglich wünschenswert, dass sich auch auf Software-Applikationsebene ein Standard durchsetzt.
Für die systemübergreifende Betrachtung von Applikationsnetzen, die heute zumeist über das TCP/IP-Protokoll kommunizieren, ist es innerhalb einzelner Netze denkbar, anhand von Header-Informationen wie Source- und Destination-IP, Port und SYN/FIN-Flags, implementierungsunabhängig Abhängigkeiten in Anwendungsnetzen zu erkennen. Damit läßt sich ein Monitoring über die Veränderungen des Kommunikationsbedarfs im Applikationsnetz betreiben, das für eine Impact-Analyse auf der Ebene von Softwareanwendungen hilfreiche und zeitnahe Daten zur Verfügung stellen kann.
Für die inaktiven, nicht erfassbaren IT-Objekte, die in den Fokus des Configuration Managements einbezogen werden, kann die Erfassung über entsprechende Identifikatoren automatisiert werden. So ist es beispielsweise im Bereich des Kabelmanagements hilfreich, über Barcodes, zukünftig möglicherweise auch RFID-Transponder, die Erfassung und Zuordnung von Kabeln zu angeschlossenen Netzwerkkomponenten zu vereinfachen.
Für IT-Anwender gilt, dass deren automatische Erfassung - sofern datenschutzrechtlich vereinbar - über den Zugriff auf IT-Ressourcen vorgenommen werden kann. Gerade für derart dynamische Konfigurationen wie die temporäre Nutzung von Ressourcen, ist die automatische Erfassung eine Voraussetzung für die Umsetzbarkeit eines erweiterten Configuration Managements. Der ITIL- und MOF-Gedanke der ausschließlich kontrollierten Änderungen an CMDB-Einträgen aufgrund eines koordinierten Change-Managements, wird für dynamische Konfigurationen (beispielsweise die wahlweise Einbuchung von mobilen Geräten in unterschiedlichen Netzen) in der Praxis unterlaufen. Stattdessen werden von Anwendern und Administratoren Systeme bevorzugt, welche die einmalige Konfigurationsänderung mit automatischer Propagierung an sonstige Konfigurationsspeicher ermöglichen. Am LRZ ist ein Administrator dafür verantwortlich, eigene Änderungen an Komponenten in den entsprechenden CI-Datensätzen nachzutragen [LRZ04] - ein kontrolliertes Change-Management mit der Überwachung des Change-Impacts wird also von den Administratoren nach eigenem Ermessen durchgeführt. Dieses Verfahren ist zwar flexibel, allerdings auch fehleranfällig bezüglich der Qualität der CMDB-Daten. Um versäumte CI-Aktualisierungen zu vermeiden, gibt es Softwarelösungen, welche in regelmäßigen Abständen auf Betriebssystemebene Prüfsummen von Konfigurationen erfassen und bei Erkennung einer Abweichung entsprechende Meldungen an das Configuration Management zur Nachtragung von Änderungen senden. Automatisierung kann also auf vielfältige Weise zur Vereinfachung, Beschleunigung und Lieferung präziser Daten an das Configuration Management beitragen.

6.3.2 CMDB-Changes

ITIL-Berater geben unterschiedliche Aussagen zur Koordination von Changes und die Aktualisierung der Daten in der CMDB ab. Die ITIL schreibt vor, dass Changes nur durch autorisierte Change-Manager durchgeführt werden dürfen. Ist der Kreis der Change-Manager jedoch zu gering gewählt, verzögern deren dokumentierenden Tätigkeiten die Arbeitsprozesse der abhängigen IT-Mitarbeiter. Im oben genannten Beispiel des LRZ wurde deshalb die Autorisierung, je nach Zuständigkeitsbereich, den ausführenden Mitarbeitern übertragen. Für Changes mit geringen Auswirkungen scheint dieses Verfahren wirtschaftlich effektiver zu sein als eine zentralistische Change-Koordination. Changes, die insbesondere CIs betreffen, die am Ende von Abhängigkeitsketten stehen, können durchaus ohne Abstimmung geändert werden. Beispielsweise ist die Änderung einer URL auf eine angepasste Dokumentation genauso wenig bedeutsam wie der kurzfristige Austausch defekter Peripheriegeräte mit nachträglicher Eintragung im Configuration Management. In der Praxis scheint es also sinnvoll zu sein, Regeln zu vereinbaren, welche unabhängige Änderungen von zu koordinierenden Changes voneinander unterscheiden.
In Abbildung 6-6 wird ein koordinierter Change-Prozess dargestellt. Auf der linken Seite befinden sich aus dem vorgeschlagenen Zustandsübergangsmodell (5-16) die jeweiligen CI-Stati im Laufe des Änderungsprozesses. An das Change-Managements wird ein RFC (Request for Change) mit den vorher festgestellten, betroffenen CIs (aus dem Incident- oder Problem-Management) heran getragen. Das Change-Management stellt an das Configuration Management eine Anfrage zur **Änderung** gewisser CIs, die seitens des Configuration Managements **geprüft** wird. Sind die CI-Änderungen unzulässig, lehnt das Configuration Management die Änderung mit entsprechenden Hinweisen ab. Werden die CI-Änderungen akzeptiert, so wird eine Autorisierung des Changes erteilt und der Zustand der entsprechenden CIs gegebenenfalls angepasst (beispielsweise 'CI in Änderung', 'wird getestet' oder 'ist produktiv'). Um in der CMDB eine Angabe über die Gewichtung eines CI-Changes aufzunehmen, kann eine vom Change Management vorgeschlagene **Klassifikation eingetragen** (1(c)vi) werden. Innerhalb des Change- und Release-Management erfolgt folglich die eigentliche Änderungsarbeit an der IT-Infrastruktur. ITIL und MOF unterscheiden zwischen Releases, also einer Sammlung von **Changes, die über das Release-Mangement koordiniert** werden, und der unmittelbaren Durchführung einzelner beziehungsweise geringfügiger Änderungen durch das Change-Management. In beiden Fällen erfolgt nach dem **Rollout** der Änderung die **Status-Rückgabe** der betroffenen CIs an das Configuration Management, welches die angegebenen Änderungen an den entsprechenden Komponenten überprüft (**Change-Review**). Sind die Änderungen zutreffend, können Sie in der CMDB übernommen und der CI-Status auf produktiv (im Produktivsystem) oder 'im Test' (im Testsystem) gesetzt werden. Gibt es Unterschiede, werden diese wenn möglich direkt **korrigiert** oder ein **Incident generiert** (4c), der die jeweils betroffenen ITSM Prozesse über eine notwendige Korrektur informiert. Sofern ein KPI auf die Datengüte der CI-Änderungen gesetzt ist, wird der Fehler für den **CI-Mängel-Report** (5) protokolliert.

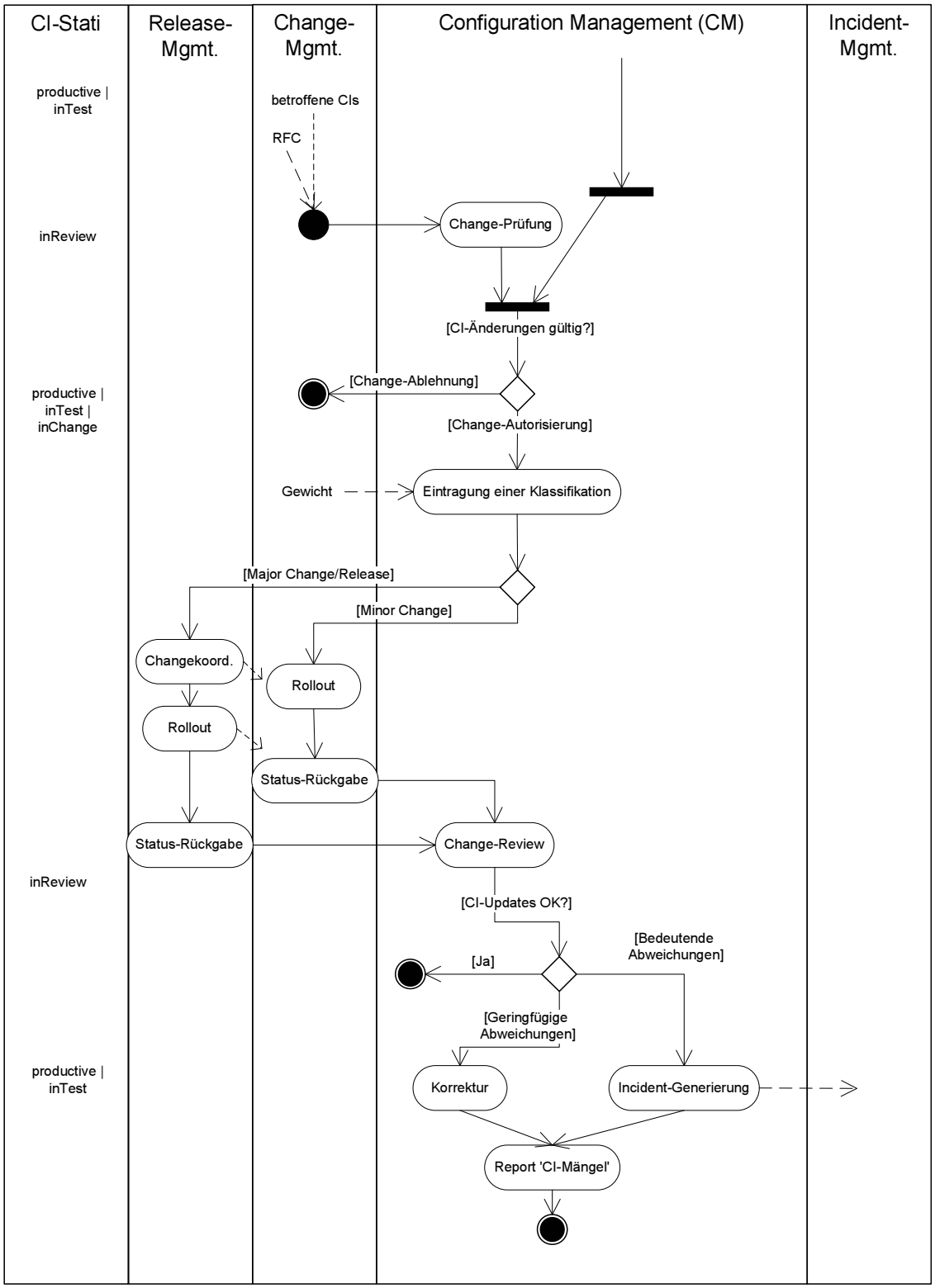


Abbildung 6-6: Arbeitsprozess CMDB-Change

6.3.3 CMDB-Abfragen

Der Abfrageprozess in Abbildung 6-7 wird von einem der ITSM Prozesse mit einem gewissen Bedarf an Konfigurationsinformationen initiiert. Die Vielfalt möglicher Abfragen wurde unter 5.2.1 und 5.6.4.7 diskutiert. Die Abarbeitung läuft auf der Seite des Configuration Managements im Idealfall automatisch ab. Nach Entgegennahme der **Abfrage** wird die **Zugriffsberechtigung** des anfragenden Benutzers, beispielsweise gegen CI-Typen, CI-Owner oder Abhängigkeiten von benutzereigenen CIs, geprüft. Ist der Zugriff erlaubt, werden die abgefragten CI-Informationen zurückgegeben, sofern vorhanden. Falls nicht vorhanden, wird der offensichtlich gegebene Bedarf an gewissen Konfigurationsinformationen protokolliert und in den **CI-Mängel-Report** aufgenommen.

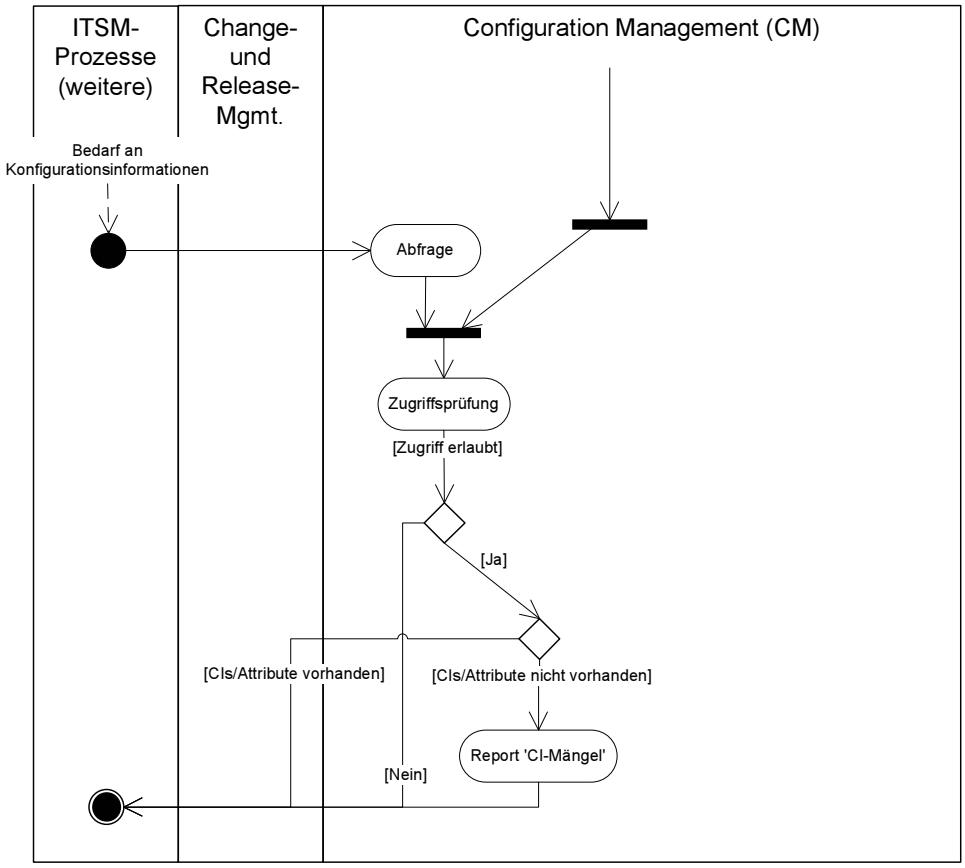


Abbildung 6-7: Arbeitsprozess CMDB-Abfragen

6.3.4 Audit und Verifizierung

ITIL und MOF schlagen die regelmäßige Überprüfung der Infrastruktur zur Verifizierung und Korrektur der CMDB-Daten vor (auch 'Status Accounting', 4). Der Prüfvorgang im ersten Prozessabschnitt in Abbildung 6-8 bezieht sich dabei auf einzelne CIs. Anhand einer **Infrastrukturanalyse** wird der IST-Zustand eines CIs erfasst und mit dem SOLL-**Zustand in der CMDB verglichen**. Falls es geringfügige Abweichungen gibt, kann im Configuration Management eine **Korrektur** der Daten vorgenommen werden. Gibt es größere Abweichungen, wird ein **Incident** zur Korrektur durch den CI-Verantwortlichen generiert. Die Abweichungen und Übereinstimmungen von IST und SOLL werden in einer **Auswertung** zusammengefasst und der IT-Strategie übergeben. Dort wird der **Erfolg des Configuration Managements bewertet** und gegebenenfalls eine **Optimierung** angewiesen, die wiederum den Planungsprozess unter 6-4 einleitet.
In den bekannten Fällen hat sich im Praxiseinsatz die regelmäßige Überprüfung der CMDB-Einträge nicht durchgesetzt. Stattdessen regeln koordinierte Changes die Aktualisierung von CIs, wobei eine entsprechende Disziplinierung zur Datenpflege bei den verantwortlichen Mitarbeitern vorausgesetzt gesetzt wird ([LRZ04] und [Mat04]). Um Audit und Verifizierung möglichst zu beschleunigen, wird verstärkt auf Discovery-Mechanismen gesetzt, welche ausschließlich die Differenzen zwischen IST und SOLL zur manuellen Auflösung aufzeigen [Jak04].

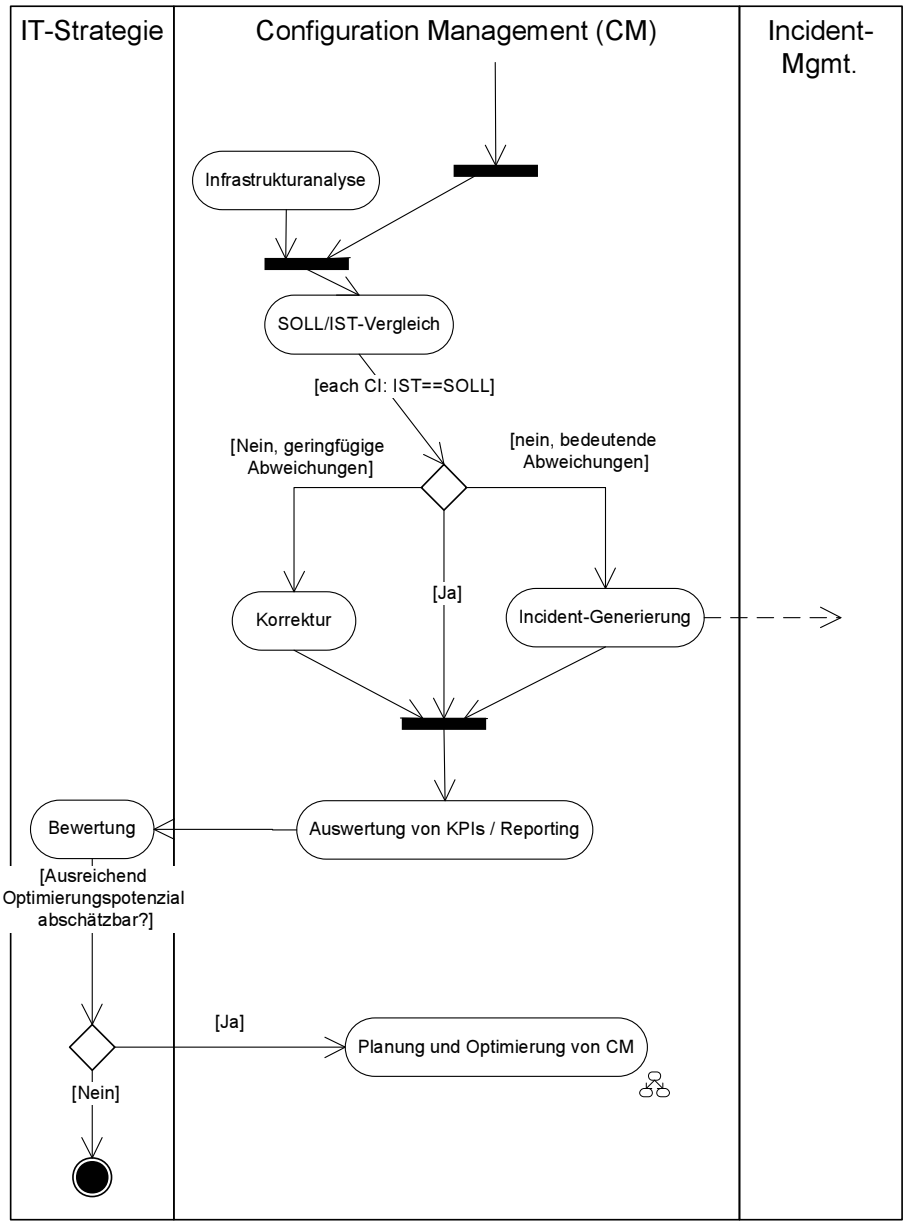


Abbildung 6-8: Arbeitsprozess Audit und Verifizierung

7 Ausblick unter Komplexitätsaspekten

Dieses Abschlusskapitel fasst die Erkenntnisse der vorhergehenden Kapitel zusammen und verschafft in Anbetracht der historischen und aktuellen Entwicklungen einen Ausblick zur anzunehmenden Fortentwicklung des Configuration Managements.

ITIL und MOF tragen nach allgemeiner Auffassung dazu bei, eine gemeinsame Terminologie des IT Service Managements und ein gemeinsames Verständnis über eine getroffene Aufteilung von ITSM Prozessen, durchzusetzen. Bezüglich der Umsetzung bleiben die beiden Frameworks allerdings unklar, so dass bei der Implementierung von ITSM Prozessen die ITIL/MOF-Literatur keine Verwendung finden kann. Als Umsetzungsvorschlag wurden in der Diplomarbeit zum einen ein Informationsmodell für die IT-Infrastrukturmodellierung nach ITIL und MOF entwickelt, das als Grundlage für ein CMDB-Design verwendet werden kann. Zum anderen wurde ein Arbeitsprozessmodell erarbeitet, das die ITIL und MOF Vorgaben zur Durchführung des Configuration Managements in einem zeitlichen und organisatorischen Rahmen strukturiert. Beides trägt durch die Konkretisierung der ITIL/MOF-Vorschläge zur beschleunigten Einführung des Configuration Managements bei.
Zum Configuration Management in der Arbeitspraxis wurde festgestellt, dass CMDB-Inhalte im gewöhnlichen dafür bereitgestellt werden, um die allgemeinen Zusammenhänge von einzelnen Systemen, darauf laufenden Anwendungen und jeweils verantwortlichem IT Personal zu dokumentieren. Die entsprechend groben Abhängigkeiten lassen keine adäquate Impact-Analyse oder Risiko-Berechnung zu, sie dienen jedoch als Ausgangsbasis für eine Experteneinschätzung. Die CMDB-Daten können als sekundäre Informationsquelle bezeichnet werden, die nicht primär, wie die IT-Infrastruktur selbst, den Wertschöpfungsprozess beeinflussen. Für derartige Informationsquellen ist die Vermittlung der Nutzung und Pflege gegenüber dem IT Personal teils erschwert, sofern der direkte Zugriff auf die IT-Infrastruktur die benötigten Daten vergleichsweise in vollem Umfang liefert. Für den Überblick über IT-Infrastrukturkomponenten wird eine CMDB jedoch im Allgemeinen als hilfreiches Instrument verwendet.
Fraglich ist, ob sich die dargestellte Rolle des Configuration Managements zukünftig verändern wird. Die folgenden Betrachtungen zur Infrastrukturentwicklung ermöglichen eine Spekulation zur zukünftigen Rolle des CM. Zum einen ist zu beobachten, dass IT in immer mehr Lebensbereichen unterstützend eingesetzt wird, insbesondere die Endgerätemobilisierung ist hierbei zu nennen. Diese Entwicklung vergrößert den Umfang einer IT-Infrastruktur und damit möglicher, zu erfassender CIs. Zum anderen ist gegenläufig zur Individualisierung und Funktionserweiterung im historischen Vergleich eine zyklische Funktionsintegration zu betrachten, die nach und nach einzelne Ebenen des Layermodells durchläuft. Beispielsweise hat sich im Laufe von 10 Jahren TCP/IP zum Standard entwickelt, der heute für die Netzwerkkommunikation meist selbstverständlich (also funktionsintegriert) genutzt wird. Die Durchsetzung von i386er Systemen hat dazu geführt, dass Rechenzentren die Vielfalt an Rechnern mit unterschiedlichen Architekturen reduziert haben. Seit einigen Jahren ist auf Hardwareebene die Integration von ehemals separaten Rechnerkomponenten in Mainboards zu betrachten. Derzeit ist die Virtualisierung ein wichtiges Schlagwort, das sich auf die Verwendung virtueller Rechner oder die Homogenisierung des Zugriffs auf verteilte Speicher im Netzwerk bezieht. In beiden Fällen wird weitergehend von der Systemebene abstrahiert und damit Funktionalität integriert. Diese Entwicklungen führen letztendlich zur Vereinfachung von Konfigurationen in den einzelnen Schichten und reduzieren die möglichen Abhängigkeiten.
Die Funktionsintegration führt zur Homogenisierung von Komponenten, die es wiederum erleichtert, Automatismen zur Erfassung von CIs anzuwenden. Wie bereits bei der Modellierung des relativ überschaubaren Szenarios festgestellt wurde, sind Automatismen zur Vereinfachung der Datenerfassung für das Configuration Management eine notwendige Voraussetzung. Bezugneh-

mend auf Abbildung 6-3, stellt sich in geringerem Maße die Frage nach dem geeigneten Umfang beziehungsweise der geeigneten Detaillierung der CI-Erfassung für ein optimales Management - stattdessen ist der Einsatz von Instrumenten wie der Automatisierung bei der Datenerfassung und der Reduzierung der Konfigurationsvielfalt ein bedeutsameres Mittel für die Aufwandsreduzierung. Die Entwicklung von Software, die im Layermodell auf Applikationsebene und allen übergeordneten Ebenen die meist dynamische Ressourcennutzung automatisch erfasst und überwacht (vgl. 6.3.1.5), ist in diesem Zusammenhang noch ein relativ unbearbeitetes Arbeitsfeld.

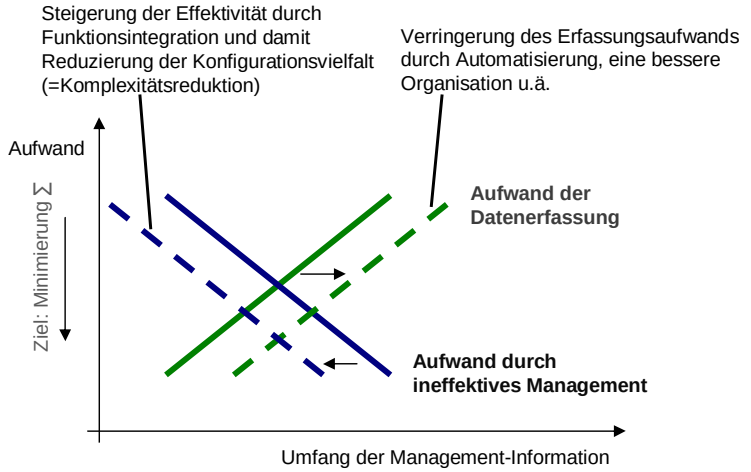


Abbildung 7-1: Aufwandsminimierung durch Automatisierung und Funktionsintegration

Von Anwenderseite wird eine über Automatismen gesteuerte, selbstpflegende Informationsbasis gewünscht. Wird auf einem Rechner beispielsweise ein neues Softwarepacket installiert, kann eine automatische Systeminventur die Änderung in der CMDB nachtragen, ohne ein aufwändiges Change-Prozedere durchlaufen zu müssen. Das entspricht einerseits nicht dem strikten Kontrollgedanken für Changes in ITIL und MOF, andererseits besteht durchaus Vereinbarkeit, sofern zulässige Changes auf unkritische CIs beschränkt werden können. Eine selbst-pflegende CMDB ist derzeit nur für homogene Infrastrukturen mit entsprechend weitgehender, integrierter Steuerung denkbar. Es erscheint wahrscheinlich, dass für die Akzeptanz eines erweiterten Configuration Mangements eine durchgängige, automatische Selbstpflege erfolgsentscheidend sein wird.

A Konfigurationsdaten nach MOF

Beschreibung im Configuration Management

Configuration management is a critical process responsible for identifying, controlling, and tracking all versions of hardware, software, documentation, processes, procedures, and all other inanimate components of the information technology (IT) organization.
Each CI can be composed of other CIs. CIs may vary widely in complexity, size, and type, from an entire system (including all hardware, software, and documentation) to a single software module or a minor hardware component [Mic04c].

Bezüge von MOF-SMFs zum Configuration Management

Optimizing Quadrant

Service Level Management

The Configuration Management SMF is critical because the CMDB allows for the service catalog components that ensure delivery of the service to be updated. In return, there should be a feed from the service catalog into the CMDB-such as when services are added as CIs to the database. This allows full impact analysis and the relationship of the components and service in the CMDB to be managed where possible [Mic03].

Capacity Management

Changes made to IT resources, also known as configuration items (CIs), and to service level objectives (SLOs) for these resources need to be reflected in the configuration management database (CMDB). Service level agreement availability and capacity data from the CMDB allows more proactive measurement of performance based on SLA compliance. This data is an important input to capacity management. Associated demand and workload requirements, resulting performance, and resource metrics are recorded in the capacity management database. Effective coordination and correlation of elements between these logical databases are required for timely information and ongoing capacity recommendation and planning [Mic04a].

Security Administration

There is a strong security component to configuration management. Configuration management deals with keeping track of the hardware owned by an organization and the versions of internal software that are used. Administrators should be aware of and in full control of the versions of the operating system, database management system, and all applications running on network computers. Poor configuration management could facilitate the introduction of malicious code into an operating system(s) or into an application [Mic02h].

Infrastructure Engineering

The Configuration Management SMF stores the IE standards and policies as CIs in the CMDB and ensures they are under the same level of control and change management as the other CIs. The CMDB is a key source of information to the IE SMF during the setup activities, and the two are used in conjunction in preparing RFCs for change authorization. The CMDB holds all

the information on the infrastructure categories, the services delivered by them, and the scope of their individual service components, so it is a valuable source of information in defining the scope and extent of the infrastructure environment.
It also facilitates, through its relational capabilities, the mapping of relationships between infrastructure categories, policies, and standards [Mic04d].

Availability Management

Availability management considers all aspects of the IT infrastructure and supporting organization that may impact availability, including training, skills, policy, process effectiveness, procedures, and tools [Mic02a].

Financial Management

Configuration management includes the processes and procedures necessary to account for the equipment in its current configuration and for the historical documentation of all subsequent changes to that configuration. Configuration management and financial management interact in several ways:

- Changes to a configuration may be costly.
- Simplifying system configurations may reduce costs.
- Accurate cost data is one tool in evaluating systems.

[Mic02c].

Workforce Management

The primary purpose of workforce management is to ensure that the correct level of staff are recruited and retained to manage the operational production environment. In addition, processes and procedures should be in place to make sure that the staffing requirements remain at the correct level. Once staff members are employed, it is the responsibility of workforce management to continuously manage the staff, skills sets, and objectives so that the correct level of staff is retained [Mic02l].

Service Continuity Management

Service continuity management primarily considers those IT assets that support key business processes. However, the installation of mechanisms to deliver service continuity management will not necessarily be sufficient to keep those business processes operating after a service disruption. Should it be necessary to relocate to an alternative working location, provisions are required for items such as office and personnel accommodations, copies of critical paper records, courier services, and telephone facilities to communicate with customers and third parties. The service continuity management process identifies the required and agreed minimum level of business operation following a service disruption, along with a requirements definition covering systems, facilities, and service requirements. The process then examines the risks and threats to these requirements and develops an IT risk reduction or mitigation program. This program implements mechanisms delivering the continuity requirements necessary to provide the required optimum level of business operation [Mic02i].

Changing Quadrant

Change Management

The goal of the Change Management SMF is to provide a disciplined process for introducing required changes into the IT environment with minimal disruption to ongoing operations.

Configuration Management provides a managed database (CMDB) for the change logs, RFCs, definitive software library (DSL), definitive hardware store (DHS), release package, and all CIs [Mic04b].

Release Management

Configuration Management provides a managed database (CMDB) for the change logs, RFCs, definitive software library (DSL), definitive hardware store (DHS), release package, and all CIs. It needs release management to update the CMDB with the release package after deployment [Mic04f].

Operating Quadrant

Directory Services Administration

Configuration management deals with keeping track of the versions of internal software that are used. It is important for administrators to clearly understand and be in full control of which versions of the operating system, database management system, and all applications are running on network machines. With respect to directory services, configuration management controls specifically which version of the directory is running, which versions of directory-enabled applications are deployed, and which versions of support, custom-built, or third-party tools are running [Mic02b].

Job Scheduling

Configuration management is the process of tracking and accounting for hardware, software, documentation, and all other components of the IT environment. Configuration management is responsible for maintaining the configuration management database (CMDB), which is used to track all IT-related components. All changes that are implemented in the batch architecture environment must be recorded in the CMDB [Mic02e].

Service Monitoring and Control

The tools available to the service monitoring and control process may be used to gather data on the physical state of configuration items (CIs) and validate the integrity of the configuration management database. (For example, do the CIs really exist? Are there CIs in production environments that are not recorded in the CMDB?) Monitoring and control could prove vital to the configuration management process to help ensure that the configuration management database is accurate. If it is not accurate, the CMDB is of little value to the other processes that make considerable use of it, such as incident management, problem management, release management, and change management.
Monitoring the IT infrastructure in the production environment should not only detect planned changes to configuration items, but also should detect unplanned changes to the environment. These unplanned changes can result in discrepancies between what is reported in the CMDB and what really exists in the IT environment. Configuration management should also work closely with release management to ensure that new service monitoring and control infrastructure, tools, and configuration changes are captured upon deployment [Mic04g].

Print and Output Management

Configuration management is an IT process used to specify, track, and report on each IT component under configuration control, referred to as configuration items (CIs). Data is stored in a logical entity known as the configuration management database (CMDB), which typically consists of multiple distinct databases. Print and output management is related to configuration management through the CMDB entries that must be processed every time a change is initiated (through change management) to any print and output management CIs (for example, printers). System

administrators and the configuration manager (owner of the configuration management process) need to come to an agreement regarding the print and output management CMDB structures (that is, attributes and relationships) for print and output management CIs (for example, hardware, software, network components, users, and so on). Note that no changes should occur to any print and output management CIs without a request for change (RFC) being processed and approved. System administrators may have to interact with various configuration domain coordinators who are responsible for various aspects of the CMDB. For example, one or more domain coordinators may be responsible for tracking different print and output management infrastructure components, such as the network, the associated printers, and so on [Mic02f].

Storage Management

Configuration management is an IT process used to specify, track, and report on each IT component under configuration control or configuration item (CI). Data are stored in a logical entity known as the configuration management database (CMDB) typically consisting of multiple distinct databases.
Storage management is related to configuration management through the CMDB entries that must be processed every time there is a change initiated (via change management) to any of the storage management configuration items. The storage manager and the configuration manager (the configuration management process owner) need to agree on the storage management CMDB structures (attributes and relationships) for storage CIs. These are hardware, software, network components, users, and so on. Note that no changes should occur to any storage management CIs without an RFC being processed and approved. The storage manager may have to interact with various configuration domain coordinators responsible for various aspects of the CMDB. For example, one or more domain coordinators may be responsible for tracking different storage management infrastructure components, such as the network, the associated disk drives, and so on [Mic02k].

Supporting Quadrant

Service Desk

The service desk function uses the information from the configuration management database (CMDB), when performing many of ist processes.
When calls are initially received, customer details may be acquired from within the CMDB. The service desk function checks these details with the customer to ensure that records are accurately maintained. When processing incidents and service requests, the service desk function checks the configuration item (CI) details against those held within the CMDB and notifies configuration management where discrepancies are found so that they can correct the errors and investigate why the discrepancies arose. The service desk function also utilizes the information within the CMDB to accurately target the supply of proactive communications to customers [Mic02j].

Incident Management

Configuration management provides vital information that is used throughout the incident management process. The configuration management database (CMDB) contains information that can be used to:

- Provide and check caller details.
- Provide information on configuration items (CIs).
- Assist with the classification of incidents by indicating services and SLAs impacted by the failure of particular CIs.
- Identify the relationship and dependencies between CIs.
- Identify identical or similar CIs for comparison purposes.

- Identify alternative routes and workarounds.
- Record changes to configuration items because of RFCs.

[Mic02d].

Problem Management

Configuration management provides vital information that is used during the problem management process. The CMDB contains information used to:

- Provide information on CIs.
- Assist with the classification of problems and known errors by indicating services and SLAs impacted by the failure of particular CIs.
- Identify the relationship and dependencies between CIs.
- Identify identical or similar CIs for comparison purposes and to prevent problem replication.
- Identify alternative routes and workarounds.
- Record changes to configuration items as a result of RFCs.

[Mic02g].

B Glossar

ARIS Architektur integrierter Informationssysteme

CI Configuration Item

CM Configuration Management

CMDB Configuration Management Database

FSC Zeitplan der geplanten Änderungen (Forward Schedule of Change)

ICT Information Communication Technology

KPI Key Performance Indicator: Schlüsselkennzahl zur Messung der Zielerreichung

OGC Office of Government Commerce

OLA Betriebsvereinbarung (Operation Level Agreement)

Prozessmodell Eine abstrakte Beschreibung eines bestimmten Prozesses. Diese Beschreibung kann mehr oder weniger formal sein. Ein Prozessmodell stellt aufgrund seiner Abstraktionsstufe immer eine bestimmte Sicht auf den beschriebenen Prozess dar [1]

RFC Request for Change

SLA Service Level Agreement: Formelle Vereinbarung zwischen Kunde und IT Dienstleister über Umfang und Güte des zu erbringenden Service [Vog02].

SMF Service Management Function, wird im MOF verwendet, enspricht nach ITIL-Terminologie einem ITSM Prozess.

UC Lieferantenvertrag (Underpinning Contract)

[1] http://swt.cs.tu-berlin.de/ pirol/glossar/Prozessmodell.html

Abbildungsverzeichnis

Literaturverzeichnis

[All04] Allrutz, Dr. Ralf: *Organic Computing, Computer- und Systemarchitektur im Jahr 2010.* Technischer Bericht, VDE/ITG/GI-Positionspapier, 2004.

[AS05] Adelsberger, Prof. Dr. Heimo H. und Markus Stallkamp: *Vorlesung Unternehmensmodellierung.* http://vawi74.wi-inf.uni-essen.de/ws0405umo/, 2005.

[BRS95] Becker, Jörg, Michael Rosemann und Reinhard Schütte: *Grundsätze ordnungsgemäßer Modellierung.* Wirtschaftsinformatik, 37:435–445, 1995.

[BSIG00] Becker, Jörg, Reinhard Schütte, Hartmut Ibershoff und Thomas Geib: *Grundsätze ordnungsgemäßer Modellierung (GoM) - Sachbericht.* Technischer Bericht, Institut für Wirtschaftsinformatik, Westfälische Wilhelms-Universtität Münster, IDS Scheer AG, Bremke & Hörster GmbH & Co., 2000.

[CD99] Clark, James und Steve DeRose: *XML Path Language (XPATH).* http://www.w3.org/TR/xpath, 1999.

[CT04] Chiu, David und D.L. Tsui: *Modeling The Enterprise IT Architecture - An IT Service Management Approach.* Technischer Bericht, BMO Financial Group, 2004.

[DKW04] Dollinger, Bernd F., Gunter Kröber und Volker Wiesinger: *ITIL-Syllabus.* http://www.dv-werk.de/fw_itil/syllabus/index.html, 2004.

[DMT04] DMTF: *Common Information Model (CIM) Standards.* http://www.dmtf.org/standards/cim/, 2004.

[FL04] Fettke, Petter und Peter Loos: *Referenzmodellierungsforschung.* Wirtschaftsinformatik, 46:331–340, 2004.

[Fra04] Frank, Ulrich: *E-MEMO: Referenzmodelle zur ökonomischen Realisierung leistungsfähiger Infrastrukturen für Electronic Commerce.* Wirtschaftsinformatik, 46:373–381, 2004.

[Fry04] Fry, Malcolm: *Remedy Customer Practices - Managing Change in the Real World.* Technischer Bericht, BMC Software, 2004.

[GNU05] GNU: *ARIS.* http://de.wikipedia.org/wiki/ARIS, 2005.

[HAN99] Hegering, Heinz-Gerd, Sebastian Abeck und Bernhard Neumair: *Integriertes Management vernetzter Systeme.* dpunkt-Verlag, 1999.

[HZB04] Hochstein, Axel, Rüdiger Zarnekow und Walter Brenner: *ITIL als Common-Practice-Referenzmodell für das IT-Service-Management - Formale Beurteilung und Implikationen für die Praxis.* Wirtschaftsinformatik, 46:382–389, 2004.

[Jak04] Jakobs, Dr. Roger: *Roadshow: Advanced ITIL-CMDB AixBOMS.* Technischer Bericht, ComConsult GmbH, 2004.

[Kai99] Kaiser, T.: *Methodik zur Bestimmung der Verfügbarkeit von verteilten anwendungsorientierten Diensten.* Doktorarbeit, April 1999.

[Knu04] KNUBLAUCH, HOLGER: *Protégé OWL Plugin - Ontology Editor for the Semantic Web.* Technischer Bericht, Stanford Medical Informatics, 2004.

[Loo04] LOONEY, B. CARTER: *COC N-Tuition Business Solutions AG.* Gespräch auf der Systems 2004, 10 2004.

[LRZ04] LRZ: *Besprechung mit Frau Dreo-Rodosek.* 25.10.2004 am Leibnitz-Rechenzentrum, 2004.

[Mag97] MAGEE, F.: *Quint Wellington Redwood's IPW: Processes for IT.* Technischer Bericht, Gartner, Inc., 1997. Note Number: P-800-228.

[Mat04] MATERNA: *ITSM-Framework-Event der Materna GmbH.* 17.11.2004, München, 2004.

[Mei00] MEISE, VOLKER: *Ordnungsrahmen zur prozessorientierten Organisationsgestaltung: Modelle für das Management komplexer Reorganisationsprojekte.* Doktorarbeit, Universität Münster, 2000.

[MH04] MILLER, ERIC und JIM HENDLER: *Web Ontology Language (OWL).* http://www.w3.org/2004/OWL/, 2004.

[Mic02a] MICROSOFT: *Availability Management - Service Management Function.* Microsoft Corporation, 2002.

[Mic02b] MICROSOFT: *Directory Services Administration - Service Management Function.* Microsoft Corporation, 2002.

[Mic02c] MICROSOFT: *Financial Management - Service Management Function.* Microsoft Corporation, 2002.

[Mic02d] MICROSOFT: *Incident Management - Service Management Function.* Microsoft Corporation, 2002.

[Mic02e] MICROSOFT: *Job Scheduling - Service Management Function.* Microsoft Corporation, 2002.

[Mic02f] MICROSOFT: *Print and Output Management - Service Management Function.* Microsoft Corporation, 2002.

[Mic02g] MICROSOFT: *Problem Management - Service Management Function.* Microsoft Corporation, 2002.

[Mic02h] MICROSOFT: *Security Administration - Service Management Function.* Microsoft Corporation, 2002.

[Mic02i] MICROSOFT: *Service Continuity Management - Service Management Function.* Microsoft Corporation, 2002.

[Mic02j] MICROSOFT: *Service Desk - Service Management Function.* Microsoft Corporation, 2002.

[Mic02k] MICROSOFT: *Storage Management - Service Management Function.* Microsoft Corporation, 2002.

[Mic02l] MICROSOFT: *Workforce Management - Service Management Function.* Microsoft Corporation, 2002.

[Mic03] MICROSOFT: *Service Level Management - Service Management Function.* Microsoft Corporation, 2003.

[Mic04a] MICROSOFT: *Capacity Management - Service Management Function.* Microsoft Corporation, 2004.

[Mic04b] MICROSOFT: *Change Management - Service Management Function.* Microsoft Corporation, 2004.

[Mic04c] MICROSOFT: *Configuration Management - Service Management Function.* Microsoft Corporation, 2004.

[Mic04d] MICROSOFT: *Infrastructure Engineering - Service Management Function.* Microsoft Corporation, 2004.

[Mic04e] MICROSOFT: *Microsoft Operations Framework, Version 3.0.* http://www.microsoft.com/mof/, 2004.

[Mic04f] MICROSOFT: *Release Management - Service Management Function.* Microsoft Corporation, 2004.

[Mic04g] MICROSOFT: *Service Monitoring and Control - Service Management Function.* Microsoft Corporation, 2004.

[MOF03] MOF: *MOF Self-Assessment Tool.* http://www.microsoft.com/.../mof/moftool.mspx, 2003.

[Nic04] NICKEL, HOLGER: *Unternehmensbefragung 'EDV-gestützte Planung und Verwaltung heterogener technischer Netzwerkinfrastrukturen'.* Technischer Bericht, ComConsult GmbH, 2004.

[NM03] NUSSDORFER, RICHARD und DR. WOLFGANG MARTIN: *RTE - Real-time-orientierte IT-Architektur.* Technischer Bericht, CSA Consulting, S.A.R.L. Martin, 2003.

[OGC00] OGC: *ITIL Service Support.* The Stationery Office Books, 2000.

[OGC01] OGC: *ITIL Service Delivery.* The Stationery Office Books, 2001.

[OGC02a] OGC: *ITIL ICT Infrastructure Management.* The Stationery Office Books, 2002.

[OGC02b] OGC: *ITIL Planning to Implement Service Management.* The Stationery Office Books, 2002.

[OGC03] OGC: *Best Practice - Questionnaire - Configuration Management.* Technischer Bericht, itSMF, 2003.

[PE01] PINK-ELEPHANT: *Mandatory, Integration and Functional Criteria for Configuration Management.* http://www.pinkelephant.com/.../Configuration_Criteria.pdf, 2001.

[PQ03] PULTORAK, DAVE und PETE QUAGLIARIELLO: *Das MOF-Taschenbuch.* Microsoft Corporation, 2003.

[Saa03] SAALMANN, ARMIN: *Referenzprofil IT Configuration Coordinator.* Fraunhofer Institut Software- und Systemtechnik, 2003.

[Sch98] SCHÜTTE, REINHARD: *Grundsätze ordnungsgemäßer Referenzmodellierung - Konstruktion konfigurations- und anpassungsorientierter Modelle.* Gabler, 1998.

[Sch03] SCHOENWAELDER, J.: *Overview of the 2002 IAB Network Management Workshop.* Technischer Bericht, IETF, 2003.

[Sie04] SIEDERSLEBEN, JOHANNES: *Moderne Softwarearchitektur.* dpunkt.Verlag, 2004.

[Sma04] *The inCharge Common Information Model.* Technischer Bericht, Smarts GmbH, 2004.

[Som04] SOMMER, JOCHEN: *IT-Servicemanagement mit ITIL und MOF.* mitp-Verlag, 2004.

[SUN01] *Suntone Architecture Methodology - A 3-dimensional Approach to Architectural Design.* Technischer Bericht, SUN Microsystems, 2001.

[Vog02] VOGT, WALTER: *fIT for benefit.* Perseo Consult AG, 2002.

[Vog04] VOGT, WALTER: *Pocket Cards IT Infrastructure Library.* Perseo Consult AG, 2004.

[Zac80] ZACHMANN, JOHN: *Zachmann Framework.* http://www.zifa.com/framework.pdf, 1980.