Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Fakultät für **Informatik**

Institut für Telematik
Cooperation & Management
Prof. Dr. Sebastian Abeck

Aspekte einer Architektur für serviceorientiertes IT-Management

Teamstudienarbeit
von

Jörg Mehlitz, Stephan Oehlert

Verantwortlicher Betreuer:	Prof. Dr. Sebastian Abeck
Betreuender Mitarbeiter:	Dipl.-Math. Klaus Scheibenberger
	Cand. Inf. Ingo Pansa

Bearbeitungszeit: 14. Februar 2008 – 14. Mai 2008

Universität Karlsruhe (TH) · Institut für Telematik · Geb. 20.20 · Zirkel 2 · 76131 Karlsruhe

Ehrenwörtliche Erklärung

Wir erklären hiermit, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben.

Karlsruhe, den 14.05.2008

Jörg Mehlitz

Stephan Oehlert

Inhaltsverzeichnis

1 EINLEITUNG

1.1 Einführung in das Themengebiet

Heutzutage wird in der Wirtschaftswelt ein Großteil der Geschäftsprozesse durch Informationstechnologie unterstützt. Abläufe werden dadurch beschleunigt und vereinfacht und Prozesse hoher Komplexität überhaupt erst ermöglicht. Die Ansprüche an IT-Unterstützung sind dadurch stark gestiegen; damit geht eine höhere Komplexität der IT-Infrastruktur einher.
Die zusätzliche Abstraktionsebene *IT-Dienst* dient als Bindeglied zwischen dem *Dienstanbieter* und dem *Dienstkunden*. Nach Huppertz [Hup06] kann ein IT-Dienst als ein Bündel von Nutzeffekten betrachtet werden, das die Perspektive des Dienstkunden wiederspiegelt. Der Dienstkunde erwartet von einem Dienst einen Beitrag zur geschäftlichen Wertschöpfung. Der *Dienstnutzer* ruft einen Dienst ab, ohne sich Kenntnisse über die konkrete Umsetzung innerhalb der IT-Infrastruktur aneignen zu müssen. Die sich ergebende *Dienstleistungsbeziehung* zwischen *Dienstanbieter* und *Dienstkunde* wird für einen konkreten IT-Dienst in einer *Dienstleistungsvereinbarung (Service Level Agreement, SLA)* festgelegt. Darin wird in einer für den Kunden verständlichen Weise festgeschrieben, welche funktionalen und nichtfunktionalen Merkmale (*Attribute*) den Dienst ausmachen. Der Dienstanbieter stellt den so vereinbarten Dienst zu Verfügung. Er trägt die Verantwortung, seine Infrastruktur so zu gestalten, dass er die zugesicherten Merkmale gewährleisten kann.
Um die Nutzeffekte eines Dienstes besser herausarbeiten zu können, versetzt sich der Dienstanbieter in die Perspektive des Dienstkunden. Durch diesen Sichtwechsel wird für den Dienstanbieter deutlich, welche Eigenschaften des Dienstes für den Kunden relevant sind. Die erzielbare Wertschöpfung wird damit für den Kunden wie für den Anbieter besser wahrnehmbar.
Im weiteren Verlauf wird sowohl der Begriff *Dienst*, als auch der englische Begriff *Service* benutzt. Auch wenn man über die inhaltlichen Unterschiede beider Wörter sicher ausführlich diskutieren kann, sollen sie im Rahmen dieser Arbeit als inhaltlich äquivalent betrachtet werden.

1.1.1 IT-Infrastrukturbetreiber als Dienstanbieter

Die Aufgabe für den Betreiber einer IT-Infrastruktur in der Rolle eines Dienstanbieters besteht darin, die in einem SLA zugesicherten Ausprägungen von Diensteigenschaften zum Zeitpunkt der Diensterbringung sicherzustellen. Ausprägungen in diesem Zusammenhang bezeichnen konkrete Zusicherungen, wie beispielsweise eine Mindestverfügbarkeit von 99,95% gemittelt über ein Jahr. Da sämtliche IT-Dienste von der vom Betreiber bereitgestellten Infrastruktur erbracht werden, ist deren korrekte Funktion eine Grundvoraussetzung für eine erfolgreiche Diensterbringung. Um eine wirtschaftliche Arbeitsweise zu ermöglichen, soll die verfügbare Leistung der IT-Infrastruktur zu der zur Erbringung des IT-Dienstes benötigten verhältnismäßig dimensioniert sein. Zu niedrig angesetzte Kapazitäten gefährden die zuverlässige Diensterbringung, starke Überkapazitäten, beispielsweise durch nicht erforderliche Redundanz, sind unwirtschaftlich.
Es ist deshalb im Interesse eines Anbieters von IT-Diensten, einen Zusammenhang zwischen den zugesicherten Eigenschaften der von ihm bereitgestellten Dienste einerseits und den Parametern der für die Diensterbringung genutzten IT-Ressourcen andererseits herstellen und überprüfen zu können. Solches Wissen basiert bisher hauptsächlich auf Kenntnissen und Erfahrungswerten von Administratoren, denen die funktionalen Zusammenhänge zwischen Infrastrukturkomponenten bekannt sind, und die so in der Regel für einen reibungslosen Betrieb sorgen können.

1.1.2 SLAs und serviceorientiertes IT-Management

Dieser Ansatz ist jedoch nicht mehr hinreichend, wenn für Diensteigenschaften in SLAs Garantien gegeben werden und damit die Anforderungen an die korrekte Erbringung eines Dienstes steigen. Eine Zusicherung, beispielsweise für einen gewissen Grad der Erreichbarkeit eines Dienstes, ohne dass diese vom Dienstanbieter sichergestellt werden kann, birgt für

Diensterbringer wie Dienstkunden große Risiken; das Vertrauensverhältnis, auf der eine solche Abmachung aufbaut, wird bei Nichterfüllung der zugesicherten Eigenschaften eines Dienstes untergraben oder zerstört.
Es werden daher Konzepte gesucht, wie die Erfüllung von Zusicherungen bzgl. eines Dienstes durch den Diensterbringer sichergestellt werden kann. In SLAs beschriebene Dienstmerkmale sollen mit Eigenschaften der IT-Infrastruktur so verknüpft werden können, dass gegebene Garantien auch mit hoher Sicherheit eingehalten werden können.
Die Kenntnis der Verknüpfungen eines Dienstes mit seinen Ressourcen und der Dienstmerkmale mit den Ressourcenmerkmalen ist in vielfältiger Weise für das IT-Management relevant.

1.1.3 Asset-Management und CMDB

Unter *Asset-Management* versteht man zunächst die Verwaltung und Nachverfolgung von Anlagegütern (z.B. Hardware, Software) in Bezug auf buchhalterische Fragestellungen wie beispielsweise den Beschaffungszeitpunkt eines Geräts, vereinbarte Garantieleistungen und Abrechnungsinformationen. Die in einem *Asset-Managementsystem* gespeicherten Daten können um Konfigurations- und Verknüpfungsinformationen erweitert und Teil einer *Configuration Management Database* (CMDB) werden. Die CMDB ist einer der zentralen Aspekte des IT Service Managements.
Das *Asset-Management* ist die Management-Domäne, die sich mit der Verwaltung und Nachverfolgung von Anlagegütern befasst. Im Kontext dieser Arbeit bezieht sich dies ausschließlich auf Anlagegüter aus der IT. Daher stehen hier im Mittelpunkt vor allem Informationen über verwendete Hard- und Software: Netzwerkkomponenten, Systeme und Software.
Informationen darüber, wie diese Komponenten konfiguriert sind, und welche Verknüpfungen zwischen ihnen bestehen, sind Teil einer CMDB.

1.1.4 Fault-/ Performance-Management und Monitoring

Das *Fault-Management* beschäftigt sich mit dem Erkennen, Isolieren, Beheben und Protokollieren von auftretenden Fehlern. Das *Performance-Management* geht in gewisser Weise darüber hinaus: Seine Aufgabe ist es, das Verhalten und die Leistungsfähigkeit von IT-Komponenten zu bewerten. Es beinhaltet also das Sammeln und Analysieren von dynamischen Statusdaten, um darüber Auskunft zu geben und nötigenfalls ein Eingreifen zu ermöglichen.
Das Überwachen der Funktion von IT-Komponenten im Netzwerk und gegebenenfalls darauf laufender Software wird üblicherweise als *Monitoring* bezeichnet. Dieses bildet ganz offensichtlich eine wesentliche Grundlage, um das Fault- und Performance-Management zu unterstützen. Aus den von einem Monitoringsystem ermittelten Statusinformationen lässt sich, nach entsprechender Analyse bzw. Interpretation, zunächst die Notwendigkeit und – falls tatsächlich Bedarf besteht – die Art und Weise eines regulierenden Eingriffs in die Infrastruktur ableiten.
Eine Unterstützung bei der Interpretation der Monitoring-Statusdaten bieten die Systeme bislang jedoch nicht oder nur in rudimentären Ansätzen. Es ist die ausschließliche Aufgabe der Netzwerk- und Systemadministratoren – mit Hilfe Ihrer Erfahrung und ihres Hintergrundwissens über die Infrastruktur und die eingesetzte Software – die Statusdaten sinnvoll zu deuten und gegebenenfalls angemessen zu reagieren. Dieser Vorgang ist also vornehmlich von Menschen abhängig und daher subjektiv.
Insbesondere die konkreten Zusammenhänge zwischen Statusdaten einerseits und Verfügbarkeit und Qualität der angebotenen IT-Dienste andererseits sind durch die hohe Komplexität der zugrunde liegenden Infrastruktur praktisch nicht oder nur in Sonderfällen ohne weiteres ermittelbar. Beispielsweise kann der Ausfall oder das Fehlverhalten einer bestimmten Komponente im ungünstigsten Fall zum kompletten Ausfall eines Dienstes führen. Da dem Monitoringsystem nicht bekannt ist, welche Konsequenzen Ausfälle von Komponenten auf den Dienst haben, kann dessen Ausfall nicht signalisiert werden. Auch Entscheidungen zu Änderungen oder Aufrüstungen von Geräten oder Software mit dem Ziel, einen Dienst besser oder zuverlässiger zu erbringen, sind selten mit formalen Verfahren oder Messdaten begründet,

sonden stützen sich hauptsächlich auf die Erfahrungen der Mitarbeiter aus der täglichen Arbeit mit der Infrastruktur.
Mit einer verstärkten Ausrichtung des Monitoring auf den Dienst bzw. dessen dynamische Aspekte, ließen sich Auswirkungen von Fehlern oder Änderungen innerhalb der IT-Infrastruktur auf IT-Dienste besser abschätzen beziehungsweise deren korrektes Funktionieren besser sicherstellen.

1.1.5 Serviceorientiertes IT-Management

Asset-Management- und *Monitoringsysteme* sind grundsätzlich voneinander unabhängig, da sie ihren Ursprung in unterschiedlichen Managementdomänen haben. Im Rahmen eines IT-Managements, das den Dienst in den Mittelpunkt stellt, zeigt sich jedoch, dass Aspekte von beiden von Bedeutung sind. Informationen beider Domänen müssen gemeinsam im gleichen Kontext betrachtet und genutzt werden können. Daher wird ein domänenübergreifender Ansatz für das serviceorientierte IT-Management benötigt. Durch eine sinnvolle Verknüpfung von *Asset-Management* und Monitoring entstünde ein Mehrwert gegenüber voneinander getrennten Managementdomänen und der IT-Dienst würde in den Fokus rücken.
Als Ausgangspunkt für eine solche Verknüpfung dient das Konzept des in der *Information Technology Infrastructure Library* (ITIL) beschriebenen *Service Specification Sheet (SSS)* eines Dienstes. Die durch das SLA erzeugten Anforderungen an den Dienst aus Sicht des Service-Kunden werden hierbei den Anforderungen an die technische Infrastruktur, die den Service erbringen muss, gegenübergestellt. Es wird also der im SSS beschriebene IT-Dienst auf die vorhandenen Komponenten der Infrastruktur abgebildet.
Das *Asset-/Configuration-Management* hat seine Stärken in der Abbildung von statischen Zusammenhängen, Monitoringsysteme können dynamische Aspekte einer Infrastruktur darstellen. Da auch bei einem IT-Dienst statische und dynamische Anteile identifizierbar sind, können an dieser Stelle Aspekte eines Ansatzes für eine Verknüpfung der beiden Managementdomänen gemacht werden (siehe Abbildung 1).
Zu beachten ist, dass in dieser Arbeit der Begriff ‚serviceorientiertes IT-Management' im Sinne des *IT-Service-Managements* (ITSM) genutzt wird. Ersterer wird benutzt, da durch ihn klar ausgedrückt wird, dass hier ein am IT-Service ausgerichtetes IT-Management gemeint ist; ITSM wird dagegen in vielen Kontexten genutzt. Auch wenn die Bedeutungen an sich synonym sind, wird auf diese Weise die intendierte Bedeutung des Begriffs deutlicher.

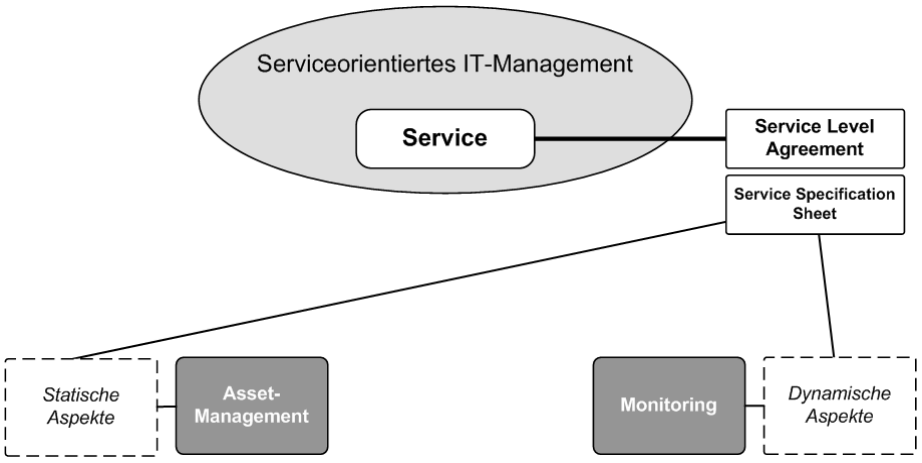

Abbildung 1: Modell eines serviceorientierten IT-Managements

1.2 Behandelte Fragestellungen

In dieser Studienarbeit werden Teile einer Architektur nach einem *Best-Practise*-Konzept erarbeitet, um die statischen und dynamischen Aspekte für das serviceorientierte IT-Management eines IT-Infrastrukturproviders bereitzustellen. Ausgehend vom SLA als Bindeglied zwischen Servicekunde und Serviceprovider sollen dafür Anknüpfungspunkte für

eine Servicedarstellung durch ein *Service Specification Sheet* für die an dieser Architektur beteiligten Managementdomänen gefunden werden. Die in dieser Arbeit betrachteten Domänen sind das *Asset-Management* und das *Fault-/Performance-Management*. Für beide wird untersucht, welche Anforderungen die Serviceorientierung mit sich bringt.
Als Grundlage für diese Untersuchung dienen Applikationen, die bereits in ihrem jeweiligen Segment etabliert sind. Für das *Asset-Management* ist dies die Software *Steinmayr NSM*, für das Fault-/Performance-Management die Software *Nagios*. Beide sollen zunächst unabhängig voneinander in den Kontext eines serviceorientierten IT-Managements gestellt und auf die dabei auftretenden Fragestellungen eingegangen werden.
Es werden Vorschläge gemacht, wie eine serviceorientierte Architektur für das IT-Management umgesetzt werden kann. Eine prototypische Umsetzung dieser Vorschläge wird vorgestellt.

1.2.1 NSM

Es soll untersucht werden ob die vorhandene Funktionalität und insbesondere das dafür genutzte Datenmodell von NSM für Einführung eines Service-Begriffs bereits mächtig genug ist; ist dies nicht der Fall, soll erarbeitet werden, wie NSM um das gewünschte Service-Management erweitert werden könnte und welche Ansätze hierfür denkbar wären.
Die Software NSM soll hinsichtlich der Möglichkeiten der Fortentwicklung hin zu serviceorientiertem IT-Management untersucht werden. Dafür wird auf einem Rechner eine Testinstallation aufgesetzt. Anhand dieser wird untersucht, wie weit die Software über reines *Asset-Management* bereits hinausgeht und wo ihre Grenzen liegen.
Die Systemumgebung, in der NSM benutzt wird, sowie Aufbau der Software werden beschrieben und es wird ermittelt, welche Bestandteile der Software im Rahmen dieser Arbeit von Relevanz sind. Für diesen Teil von NSM wird ein Überblick über die gebotene Funktionalität gegeben.
Für diese Analyse der Software wird eine Testinstallation aufgesetzt und Testdaten in diese eingefügt. Diese wird zum einen genutzt, um die Funktionalität im Rahmen des *Asset-Managements* bzw. der CMDB zu untersuchen, zum anderen, um das Datenmodell der Software zu analysieren.
Um die Arbeit mit NSM zu veranschaulichen, und um eine Grundlage für die folgende Analyse zu bilden, wird eine ansatzweise Umsetzung des Beispielszenarios ATIS Infrastruktur in Cable- und Logic-NSM vorgenommen. Für diese Umsetzung wird der grundlegende Workflow mit diesen Tools beschrieben.
Die vorhandenen Informationen über Cable- und Logic-NSM werden ausgewertet, insbesondere aus der von der Fa. Steinmayr bereitgestellten Dokumentation über die Software [Ste06a, Ste06b, Ste08]. Zusammen mit dem durch die Arbeit mit NSM erlangten Wissen über die Software werden sie genutzt, um ein für die spätere Arbeit hinreichend genaues Bild von NSM zu erlangen. Das Datenmodell von Cable- und Logic-NSM wird dahingehend analysiert, dass Assoziationen zwischen Objekten und der Zusammenhang mit der Scan-Engine deutlich werden.
Es wird eine Motivation für Aspekte eines serviceorientierten IT-Management formuliert, die für eine Software wie NSM relevant sein können. *Service Specification Sheets* werden so interpretiert, dass sie als Grundlage für eine serviceorientierte Sichtweise im Rahmen dieser IT-Management-Software dienen können.
Ein so konkretisiertes *Service Specification Sheet* dient für die Einführung eines Service-Begriffs in die Software NSM. Es wird ausgeführt, wie auf diese Weise serviceorientiertes IT-Management aussehen kann.
Die beispielhafte Umsetzung durch ein, Aspekte dieses Vorschlags aufgreifenden, neues Modul für NSM durch die Fa. Steinmayr wird beschrieben. Insbesondere wird auf darauf eingegangen, wie sich dieser Demonstrator von der vorgeschlagenen Umsetzung unterscheidet.

1.2.2 Nagios

Analog zu den Zielsetzungen bei der Software NSM soll auch für Nagios untersucht werden, inwieweit es sich in eine serviceorientierte IT-Managementarchitektur integrieren lässt.

Die Funktionalität und der Einsatzzweck der Monitoringsoftware Nagios werden beschrieben. Es wird erläutert, welchen Nutzen Nagios für den Betrieb einer Infrastruktur bringt. Um sich ein genaueres Bild von den Möglichkeiten der Software zu machen, werden der interne Aufbau, die Bestandteile und die von Nagios zur Abbildung einer Infrastruktur vorgegebenen Begriffe erläutert. Insbesondere wird auf das Informationsmodell und das Manager-Agent-Konzept von Nagios eingegangen.
Der momentane Zustand des Einsatzes von Nagios zur Unterstützung der Administration der ATIS-Infrastruktur wird beschrieben.
Es sollen verschiedene Ansätze untersucht werden, wie mit Nagios ein abstrakter IT-Service dargestellt werden kann. Dazu wird zunächst versucht, die an der Erbringung des Beispielszenario-E-Mail-Dienstes beteiligten Komponenten mit Hilfe der Möglichkeiten von Nagios zu visualisieren und damit zu einer servicebezogenen Sicht auf den E-Mail-Dienst zu kommen. Die Möglichkeiten dieser Darstellung werden untersucht, aber auch deren Einschränkungen in Bezug auf eine Serviceorientierung herausgearbeitet.
In einem weiteren Ansatz wird das Informationsmodell von Nagios genauer analysiert und Möglichkeiten untersucht, auf welche Weise es erweitert werden muss, um serviceorientiertes Monitoring besser zu unterstützen.

1.3 Beispielszenario

Zur Veranschaulichung der in dieser Arbeit behandelten Fragestellungen dient die Netzinfrastruktur der ATIS und der von ihr betriebene E-Mail-Dienst.

1.3.1 Die ATIS

Die *Abteilung Technische Infrastruktur (ATIS)* ist eine Einrichtung der Fakultät für Informatik an der Universität Karlsruhe. Sie zeichnet sich verantwortlich für den Betrieb der fakultätsinternen IT-Netzinfrastruktur und stellt den angeschlossenen Instituten eine Reihe von IT-Diensten zur Verfügung. Dazu zählen E-Mail, FTP-Server, VPN-Zugang und viele weitere. Insgesamt werden rund ein Dutzend Institute mit vielen hundert Mitarbeitern und Studenten versorgt.
Viele der IT-Dienste sind im Arbeitsalltag der Nutzer essenziell. Das Ziel der ATIS ist es deshalb, diese mit möglichst hoher Qualität und Verfügbarkeit anzubieten. Die zuständigen Administratoren werden bei dieser Aufgabe von Monitoring-Systemen unterstützt, die den Status von Infrastrukturkomponenten überwachen. Sie bieten ständig aktualisierte Informationen, die sich über ein Web-Interface von Mitarbeitern abrufen lassen. Des Weiteren können Benachrichtigungen (per E-Mail, SMS, etc.) automatisiert verschickt werden.
Zurzeit werden zwei Softwarelösungen für das Monitoring eingesetzt: Die einfache Überwachungssoftware BigBrother und seit etwa einem Jahr eine deutlich leistungsfähigere Lösung auf Basis der Software Nagios. In naher Zukunft wird BigBrother durch Nagios vollständig ersetzt werden (siehe [Pan07]).
Das IT-Management muss diese Komplexität beherrschen. Mit wachsenden Ansprüchen an IT-Services wird ein integriertes Management wichtiger. Im Mittelpunkt stehen klar definierte IT-Services als Dienstleistung für Service-Nutzer. Hohe Verfügbarkeit der Services ist notwendig. Es enthält ein Modell von aktiven Komponenten der Infrastruktur (Server, Netzwerkhardware, Software..) und wird zum Monitoring dieser Komponenten genutzt.

1.3.2 E-Mail als Beispiel eines IT-Dienstes

Diese Studienarbeit greift exemplarisch den Dienst *E-Mail* der ATIS auf. Er eignet sich gut für die Untersuchung der Fragestellungen bezüglich serviceorientierten IT-Managements, da die für die Erbringung des Dienstes notwendige Infrastruktur nicht übermäßig komplex ist. Sie ist jedoch hinreichend umfangreich, um als Beispielszenario dienen zu können. Ein weiterer Vorteil ist die weltweite Verbreitung von E-Mail, die Tatsache, dass praktisch jeder diesen Dienst benutzt und sich damit eine gute Nachvollziehbarkeit des Szenarios ergibt. Darüber hinaus existiert bereits ein großer Erfahrungsschatz bei der Administration in der ATIS.

Der hier betrachtete E-Mail-Dienst stellt den Kern-Maildienst der ATIS dar. Das bedeutet, es werden nur diejenigen Server berücksichtigt, auf denen direkt mit dem Dienst zusammenhängende Prozesse laufen. Dazu kommen die Netzinfrastrukturgeräte, also Switches und Router, die erforderlich sind, um diese Rechner untereinander zu verbinden. Außerdem zählen relevante Basisdienste wie beispielsweise DNS dazu, die für die grundlegende Kommunikation erforderlich sind. Die Beschränkung auf den Kern-Maildienst bedeutet, dass der Dienst unter Umständen auch dann als funktionsfähig gilt, auch wenn ein konkreter Nutzer, beispielsweise aufgrund von Problemen mit der externen Netzwerkanbindung, nicht erfolgreich darauf zugreifen kann. Der Fehler ist dann jedoch nicht dem E-Mail-Dienst, sondern dem zusätzlich erforderlichen Datentransportdienst anzulasten, der die Verbindung zwischen Nutzer und dem Core-Netzwerk der ATIS herstellt.
Ausführungen über die Struktur des E-Mail Dienstes der ATIS sind bereits von Pansa in [Pan07] gemacht worden. Hier wird noch einmal auf die wichtigsten Hard- und Softwarekomponenten eingegangen:
Die für den Kern-E-Mail-Dienst benötigte Infrastruktur besteht aus drei Switchen und drei physikalischen Server-Rechnern (siehe Abbildung 2). Zu den Netzwerkgeräten zählen der zentrale Core-Switch (nbi-u2-bd8810-01) und zwei direkt damit verbundene Server-Switche.

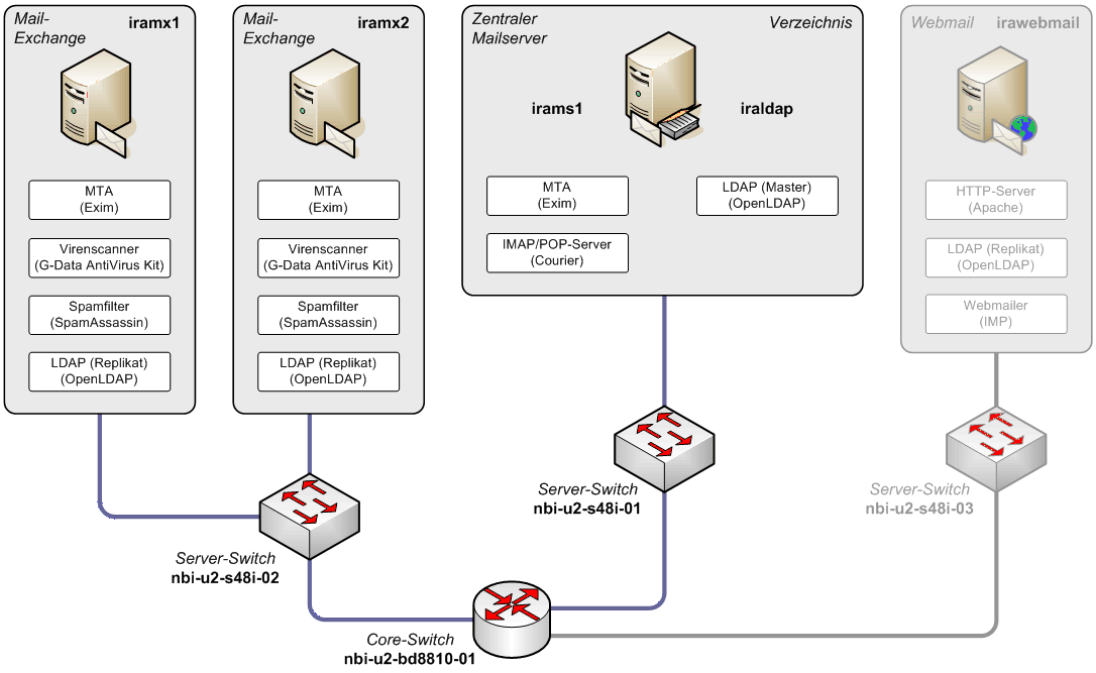

Abbildung 2: Infrastruktur für den Dienst E-Mail

An einem Server-Switch (nbi-u2-s48i-02) sind iramx1 und iramx2 angeschlossen, die als Mail-Exchanger dienen. Ihre Hauptaufgabe ist also das Weiterleiten von ein- und ausgehendem E-Mail Verkehr. Zu diesem Zweck läuft auf beiden Servern ein sogenannter Mail Transfer Agent (MTA); es wird die Open Source Software *Exim* eingesetzt. Bei einer eingehenden E-Mail wird zunächst überprüft, ob sie an eine gültige Empfängeradresse gerichtet ist. Danach wird sie an den für diese Adresse zuständigen Mailserver innerhalb des Informatik-Netzes weitergereicht. Für diese Funktionen wird ein Verzeichnisdienst benötigt, der unter anderem Informationen über die vorhandenen Benutzer, deren E-Mail-Adressen und Institutszugehörigkeit bereithält. Hierfür wird die Open Source Software OpenLDAP benutzt. Auf iramx1 und iramx2 läuft jeweils eine Replikat der zentralen LDAP-Datenbank. Eingehende E-Mails werden darüber hinaus durch die Software G-Data Antivirus auf Viren überprüft und von der Open Source Software SpamAssassin einem Spam-Scoring unterzogen. iramx1 und iramx2 sind redundant ausgelegt, d. h. sie werden im Normalfall für *Load Balancing* eingesetzt und können im Störfall die Funktion des jeweils anderen übernehmen.

An einem anderen Server-Switch (nbi-u2-s48i-01) ist ein weiterer Server angebunden. Er erfüllt mehrere Funktionen: Als iraldap läuft auf ihm die zentrale LDAP-Instanz, die von allen anderen Instanzen im ATIS-Subnetz repliziert wird. Alle Lesezugriffe, und damit der größte Teil der Anfragen an den Verzeichnisdienst, können auf diese Weise ressourcensparend und schnell von diesen Servern an ihr eigenes Replikat gestellt werden. Nur Schreibzugriffe werden ausschließlich an die zentrale Instanz weitergereicht, um einen konsistenten Datenbestand sicherzustellen. Außerdem laufen auf diesem Gerät die Prozesse des IMAP/POP-Servers irams1, der die Mailboxen der ATIS hostet und dem Endnutzer zum Abrufen der E-Mails dient. Hierfür wird die Open Source Software Courier genutzt.
In Abbildung 2 ist auf der rechten Seite noch ein weiterer Switch (nbi-u2-s48i-02) mit angeschlossenem Server zu erkennen (irawebmail), der das Webmail-Frontend hostet. Mittels eines hier installierten Apache-Webservers und der Open Source Software IMP können die Nutzer weltweit über einen Browser auf ihr E-Mail-Konto zugreifen. Zur Authentifizierung läuft auch auf diesem Server ein replizierender LDAP-Verzeichnisdienst. nbi-u2-s48i-02 und irawebmail sind zwar ohne Zweifel Teile der Mail-Infrastuktur, sind jedoch nicht essentiell für die grundsätzliche Funktionsfähigkeit des Kern-Maildienstes. Webmail bietet dem Nutzer vielmehr eine zusätzliche Möglichkeit, von extern auf die Funktionen des Kerndienstes zuzugreifen. Die beiden Geräte werden aus diesem Grund nicht zum Kern-Maildienst gezählt; sie sind daher eingegraut dargestellt und werden im Folgenden nicht mehr berücksichtigt.

1.4 Gliederung der Arbeit

Kapitel	Inhalt
1 – Einleitung	▪ Einführung in das Themengebiet IT-Management und IT-Services ▪ Einordnung der Problemstellung ▪ In der Arbeit behandelte Fragestellungen ▪ Beispielszenario E-Mail-Dienst der ATIS ▪ Gliederung der Arbeit
2 – Grundlagen	▪ Einführung in das thematische Umfeld ▪ Erläuterung zentraler Begriffe: o IT-Service o ITIL o Service Level Management o Service Level Agreement o Operation Level Agreement o Service Specification Sheet o Configuration Management Database o WBEM o CIM
3 – Analyse von Steinmayr NSM anhand eines Beispielszenarios	▪ Beschreibung von Aufbau und Struktur der Software Netz Service Management (NSM) der Firma Steinmayr Net Intelligence ▪ Analyse von Cable- und Logic-NSM anhand eines Beispielszenarios
4 – NSM im Rahmen eines serviceorientierten IT-Managements	▪ Motivation für serviceorientiertes IT-Managment mit NSM ▪ Konzept zur Modellierung eines Service Specification Sheets in NSM ▪ Vorschlag zur Umsetzung eines Service Specification Sheets durch Service-NSM ▪ Beschreibung des Demonstrators der Fa. Steinmayr
5 – Analyse von Nagios und IT-Infrastrukturmonitoring	▪ Einführung in Infrastruktur-Monitoring mit Nagios ▪ Beschreibung der Struktur und Funktionsweise von Nagios ▪
6 – Service Specification Sheet für ein Monitoringsystem	▪ Service-Baum für das Monitoring ▪ Beispielszenario E-Mail-Service im Monitoring-Service-Baum
7 – Nagios als Teil eines serviceorientierten IT-Managements	▪ Evaluation Nagios für serviceorientiertes Monitoring ▪ Darstellung des Service-Baums in einer NagVis-

	Karte ▪ Modellierung des Service-Baums in dem Informationsmodell von Nagios ▪ Anbindung des Nagios-Datenmodells an ein WBEM-System ▪ Nagios-Agenten als CIM-Provider
8 – Zusammenfassung, Fazit und Ausblick	▪ Zusammenfassung der Ergebnisse der Arbeit mit NSM und Nagios ▪ Zusammenführung des Konzeptes im Rahmen eines integrierten Managements ▪ Ausblick auf weitergehende Arbeit
9 – Anhänge	▪ Abbildungsverzeichnis ▪ Index ▪ Abkürzungen und Glossar

2 GRUNDLAGEN

In diesem Kapitel werden begriffliche Grundlagen geschaffen. Ausgehend von der Definition des IT-Service werden die im späteren Verlauf der Arbeit benötigten Begriffe erklärt. Einige Aspekte werden gesondert herausgestellt und erläutert.

2.1 IT-Service

Der Begriff *Service* wird im Bereich der Informationstechnologie in teilweise sehr unterschiedlichen inhaltlichen Zusammenhängen benutzt. Beispielsweise wird gelegentlich auch ein Systemprozess als Systemdienst oder (System-)Service bezeichnet. Hier ist jedoch etwas grundlegend anderes gemeint als beim IT-Service, auf den in dieser Arbeit Bezug genommen wird. Bei Ausnahmen von dieser Regel wird darauf hingewiesen oder die Abweichung ist durch den Kontext klar ersichtlich.
Als Grundlage dient die Beschreibung aus [CN04]. Danach ist ein IT-Service ein Bündel von Nutzeffekten, das

- durch Aktivitäten eines *Service-Providers* erbracht wird,
- durch IT- und Nicht-IT-Einrichtungen erzeugt wird,
- vom *Service-Provider* an Servicekunden verkauft wird,
- den Mitarbeitern des Servicekunden sowie anderen berechtigten Personen (Servicenutzern) bereitgestellt wird,
- von den Servicenutzern eingesetzt wird, um ihre geschäftlichen Aufgaben auszuführen bzw. zu unterstützen.

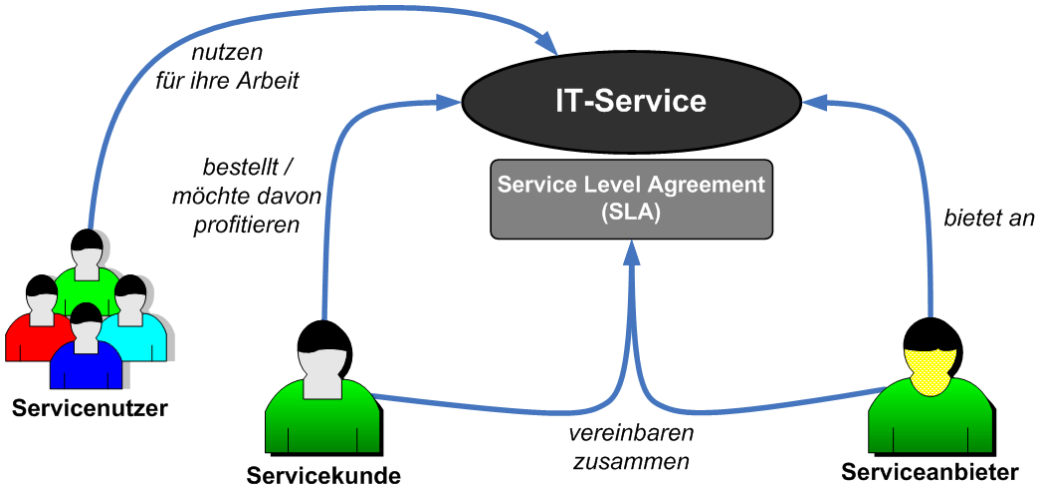

Abbildung 3: IT-Service, Rollen und Aktivitäten

Hier werden bereits die unterschiedlichen Rollen erwähnt, die die Beteiligten einer Service-Vereinbarung annehmen:

Servicekunde (Service Customer)

Der Servicekunde beauftragt den *Service Provider* mit der Bereitstellung eines Service. Er bezahlt für die Leistung, die der Service darstellt und definiert, welche Mindestanforderungen der Service erfüllen muss.

Servicenutzer (Service Consumer, User)

Der User ist der Endbenutzer, der im Rahmen seiner geschäftlichen Arbeitstätigkeit auf den Service zugreift. Qualitative Eigenschaften des Service wie Verfügbarkeitsgarantien bemerkt der Servicenutzer, indem der Zugriff auf den Service hinreichend reibungslos funktioniert.

Service-Anbieter (Service Provider)

Der Service-Anbieter ist eine Organisation oder eine Einheit davon, die im Auftrag des Servicekunden und für die Servicenutzer Dienstleistungen bereitstellt und erbringt. Er betreibt die dafür notwendige Infrastruktur und trägt die Verantwortung für die Erfüllung der im SLA vereinbarten Eigenschaften des Services [TSO01].
IT-Services umfassen im Rahmen dieser Arbeit ausschließlich IT-Systembasierte Services. Zugriff auf den Service erfolgt ausschließlich über technische Zugänge wie Netzwerk-Adressen und Ports und standardisierte Protokolle.

2.1.1 Information Technology Infrastructure Library (ITIL)

Als de-facto-Standard für das IT-Service-Management hat sich inzwischen die *Information Technology Infrastructure Library (ITIL)* durchgesetzt,. Sie bietet eine Sammlung von Best-Practise-Prozeduren und Methoden an. Die aktuelle Version ITIL V3 [ITIL08] definiert fünf Aufgabenbereiche. Bei allen Bereichen steht der IT-Service im Mittelpunkt des Interesses:

- *Service Strategy* (Strategie),
- *Service Design* (Entwurf),
- *Service Transition* (Überleitung zum Betrieb),
- *Service Operation* (Betrieb),
- *Continual Service Improvement* (Fortlaufende Verbesserung).

2.1.2 Service Level Management

Essenzieller Bestandteil des Service Managements ist das Service Level Management. Im Service Level Management werden die vom *Service-Provider* erbrachten IT-Services geplant, koordiniert, entworfen, vereinbart, überwacht und über den Status berichtet [TSO01].
Daher ist wichtiger Teil des *Service Level Managements*, eine Darstellungsform für IT-Services zu finden, die das Service-Management eines *Service-Providers* darin unterstützt, diesen zu den vereinbarten Konditionen erbringen zu können. Monitoring der IT-Services im Rahmen des *Service Level Managements* ist notwendig, um hinreichend schnell Probleme zu erkennen und beheben.
Ziel von Entwicklungen im *Service Level Management* ist es, den Grad der Integration zwischen Management-Tools zu erhöhen. Dann ließen sich Informationen, die an einer Stelle über einen Service zur Verfügung stehen, etwa an andere Stelle zur Verbesserung des Service Managements nutzen; die durch unterschiedliche Datenbasen von Management-Tools unvermeidliche Redundanz ließe sich verringern.

2.1.3 Service Level Agreement (SLA)

Die Zusicherung eines *Service-Providers*, einem Kunden einen Service mit einer gewissen Funktionalität zu einer bestimmten Qualität zu erbringen, wird in einem *Service Level Agreement* (SLA) festgelegt. Es dient als Vereinbarung zwischen Serviceanbieter und Servicekunde, um Verantwortlichkeiten und relevante Aspekte eines Service festzulegen und damit eine vertragliche Basis zwischen beiden Parteien zu bilden.
Die Struktur eines SLA ist nicht festgelegt und muss sich daran orientieren, wie sich das Verhältnis zwischen Service-Kunde und Service-Erbringer am besten beschreiben lässt. Möglich sind unterschiedliche Ansätze; ein SLA kann einheitlich für alle Service-Kunden sein, oder es können unterschiedliche SLAs für Service-Kunden-Gruppen festgelegt werden. Letzteres ist sinnvoll, wenn ein einheitliches SLA für alle Service-Kunden zu unflexibel ist und mehr Flexibilität in den Zusicherungen benötigt wird.

2.1.4 Operational Level Agreement (OLA)

Das vertragliche Verhältnis zwischen Servicekunde und Serviceanbieter in einem SLA lässt sich auch auf Prozesse innerhalb einer Organisation übertragen. In einem *Operational Level Agreement* (OLA) können Leistungen als Services eines Teils einer Organisation für andere Teile beschrieben und vereinbart werden.
Damit wird der Gedanke der Serviceorientierung nicht mehr nur für den Servicekunden ‚außerhalb' sichtbar; es können Abhängigkeiten und Verantwortlichkeiten innerhalb von Organisationen sichtbar gemacht und vertraglich vereinbart werden.

2.2 Service Specification Sheet

In einem SLA werden Eigenschaften eines Services als Vereinbarung zwischen Servicekunde und *Service-Provider* beschrieben. Für das Service-Management beim Provider ist es entscheidend, wie im SLA garantierte Eigenschaften eines Services zuverlässig erbracht werden können.
Oft werden SLAs zwar festgelegt, aber es fehlt eine hinreichend genaue Darstellung darüber, wie der im SLA festgelegte Service von der Infrastruktur letztlich erbracht wird. Dies ist aber eine Grundvorraussetzung dafür, im SLA gegebene Garantien einhalten zu können. Ohne einem SLA weitergehende Informationen zuzuordnen, was für die Erbringung notwendig ist, können Garantien nicht gegeben werden.
Service Specification Sheets stellen ein Modell von Teilen der Infrastruktur dar. Aus ihm soll hervorgehen, wie Ressourcen der Infrastruktur zur Erbringung eines Service beitragen. Damit lässt sich eine Darstellung des Service als Produkt der Zusammenarbeit zwischen Ressourcen der Infrastruktur erstellen.

2.2.1 Einordnung des Service Specification Sheets in das IT-Mangement nach ITIL

Das SSS ist Bestandteil des Service-Managements nach ITIL. Es ist eine Best-Practise-Technik für das Service-Management. Es ist eine Methode, mit der auf praktikable Weise ein Modell der zur Erbringung eines Service notwendigen Komponenten erfasst werden können.
Ein *Service Specification Sheet* ist Teil des *Configuration Managements* und kann Teil einer CMDB sein.
Ein praktisch orientierter Ansatz ist es, ein SSS für eine technisch orientierte Darstellung einer Infrastruktur hin zum integrierten Service-Management zu betrachten. Es kann damit nah am *Asset-Management* bleiben und sich an bereits eingesetzten Applikationen orientieren.

2.3 IT-Service-Management

Das IT-Service-Management umfasst diejenigen Disziplinen des IT-Managements, die einen *Service-Provider* dabei unterstützen, seine Services derart bereitzustellen, dass die in einem SLA zugesicherten Qualitätsniveaus des Service erreicht werden. Dabei wird die Gesamtheit der Methoden und Maßnahmen einbezogen, die nötig sind, um dieses Ziel zu erreichen.
Stellt man einen definierten IT-Service als Produkt einer IT-Infrastruktur in den Mittelpunkt, wird das Service-Management für diesen Service zentraler Bestandteil des IT-Managements. Der IT-Service wird die Koordinierungsgrundlage, anhand der sich das IT- Management ausrichten muss.

2.4 Configuration Management Database

Mit steigender Bedeutung von IT-Dienstleistungen einer IT-Infrastruktur wachsen auch die Ansprüche an die Zuverlässigkeit und Zusammenarbeit der IT-Komponenten selbst. Es ist essenziell, den Überblick über die Bestandteile der Infrastruktur zu behalten, da sonst die zuverlässige Aufrechterhaltung und Veränderungen erschwert oder unmöglich werden [BMC06].

Um dem gerecht zu werden, implementieren viele Organisationen eine *Configuration Management Database* (CMDB). Wichtigste Grundlage hierfür und De-Facto-Standard ist die nach ITIL definierte CMDB.

2.4.1 Struktur einer CMDB

In einer CMDB sollen alle Komponenten einer Infrastruktur erfasst, gemanagt und überprüft werden können. Sie soll damit ein umfassendes Modell einer IT-Infrastruktur ergeben. Traditionell liegen diese Informationen, sofern überhaupt vorhanden, verteilt in sehr unterschiedlichen Datenbanken. Diverse Softwareprodukte auf dem Markt erfassen unterschiedliche Aspekte einer IT-Infrastruktur.
Durch die große Uneinheitlichkeit zwischen Infrastrukturen und den vielen unterschiedlichen Herangehensweisen für die Bereitstellung von IT-Diensten beschränken sich die meisten Produkte auf einen beschränkten Teil des IT-Managements. Umfassende Lösungen sind sehr teuer und für kleinere Infrastrukturbetreiber ungeeignet.
Daher steht der Wunsch einer umfassenden CMDB einer Vielzahl beschränkter ‚Insel'-Lösungen gegenüber. Es haben sich mehrere mögliche Ansätze gebildet:

1. Vollständig isolierte Speicher

Jede der Management-Domänen behält ihr eigenes Datenmodell und ihre eigene Datenbank. Informationen über die Infrastruktur müssen immer direkt an die dafür zuständige Applikation gestellt werden. Dies erlaubt keinerlei Integration der unterschiedlichen Komponenten und stellt hohe Anforderungen an denjenigen, der die entsprechenden Informationen sucht.

2. Eine zentrale Datenbank

Alle Applikationen teilen die von ihnen verwalteten Daten einer zentralen Management-Applikation mit. Diese bildet die CMDB. Sie hat damit einen zentralen Zugangspunkt und eine einheitliche Sicht auf die Infrastruktur.
Der Nachteil einer solchen Lösung ist, dass sie hohe Anforderungen an die beteiligten Applikationen stellt. Diese müssen ihre Daten entsprechend aufbereiten, damit sie für eine zentrale Datenbank von Nutzen sind. Ohne übergreifende Standards bedeutet dies eine feste Bindung an fremde Software des Herstellers der CMDB.

3. Integrierte Datenspeicher

Jede Applikation bleibt weiterhin funktional unabhängig von anderen. Allerdings stellt sie die in ihrer Datenbank enthaltenen Informationen anderen Applikationen über definierte Schnittstellen zur Verfügung. Benötigt eine Applikation Informationen aus anderen Datenbanken, nutzt sie deren Schnittstellen, um diese zu erhalten.
Dieser Ansatz erhöht die Komplexität des Gesamtsystems beträchtlich. Er stellt wiederum höhere Anforderungen an die Infrastruktur für ein solches IT-Management. Der Nutzen besteht in einer insgesamt deutlich leistungsfähigeren CMDB und einem höheren Grad an Integration der verschiedenen Management-Bestandteile.

2.4.2 Ziele einer CMDB

Das Konzept einer CMDB ist entstanden aus dem Bedürfnis, einen einheitlichen Zugangspunkt zu allen relevanten Informationen über eine IT-Infrastruktur zu erhalten [BMC06]. Im Gegensatz zu spezialisierten Datenbanken für *Asset-Management*, SLAs, Performance Management usw. sollen alle Daten in einer einzelnen zentralen Datenbank gespeichert werden; sie dient dann als Quelle für sämtliche Anwendungen, die Zugriff auf entsprechende Teile der Daten benötigen. Alle Teile der IT-Infrastruktur werden als *Configuration Items* (CIs) in der CMDB abgelegt und verwaltet.

Ziele einer CMDB sind somit:

- Erfassung aller IT-Vermögenswerte und deren statische Konfiguration

- Bereitstellung dieser Informationen zur Unterstützung des übrigen IT-Managements
- Erkennung und Beseitigung von Abweichungen zwischen Infrastruktur und in der CMDB abgelegten Daten

Werden diese Ziele erreicht, kann dies die Managementfähigkeit einer Infrastruktur verbessern. Änderungen an der Infrastruktur können besser verfolgt werden und Konflikte leichter erkannt werden. Damit können Probleme besser gefunden und Fehler schneller beseitigt werden.
Zusammenhänge werden durch eine CMDB leichter verständlich und die Komplexität sinkt.

2.5 Web-Based Enterprise Management (WBEM)

2.5.1 Entstehung / Historie

Das IT-Management muss ich an die Gegebenheiten der rasch steigenden Komplexität und größeren Heterogenität heutiger IT-Infrastukturen anpassen. Bereits im Jahr 1996 wurde zu diesem Zweck von den Firmen BMC Software, Cisco Systems, Compaq Computer, Intel und Microsoft die *Web-Based Enterprise Management (WBEM)* Initiative ins Leben gerufen. Deren Ziel war und ist es, eine offene, herstellerneutrale Architektur für das einheitliche webbasierte Management einer verteilten IT-Infrastruktur zu entwickeln.

2.5.2 Ziele

Zwei Aspekte stehen bei WBEM im Vordergrund: Mit Hilfe eines umfassenden Informationsmodells soll die Darstellung von Managementinformationen standardisiert werden. Nur so kann ein verlustfreier und uneingeschränkter Informationsaustausch gewährleistet werden. Die *Distributed Management Task Force (DMTF)* beschäftigt sich aus diesem Grund mit der Entwicklung eines standardisierten Informationsmodells namens *Common Information Model (CIM)*. Es handelt sich um ein vollständig objektorientiertes Schema zur Beschreibung von für das Management relevanten Objekten eines Systems. Objektorientierte Eigenschaften wie Assoziationen und Vererbung werden unterstützt, darüber hinaus ist CIM unabhängig von einer bestimmten Programmiersprache.
Das zweite Ziel ist die Bereitstellung einer standardisierten Methode, um auf Managementinformationen zugreifen zu können. Damit soll der Nachteil umgangen werden, dass bisher in einer heterogenen IT-Infrastruktur für jede Managementumgebung proprietäre Software oder APIs benutzt werden müssen, um auf Managementdaten zugreifen zu können. [Heid03]

2.5.3 Aufbau

Zentrales Element der WBEM-Architektur ist der WBEM/CIM-Server (siehe Abbildung 4). Seine primäre Aufgabe besteht in der Vermittlung zwischen den CIM-Clients, die im Auftrag der mit dem Management betrauten Personen oder übergeordneten Managementsystemen (Operatoren) arbeiten, und den CIM-Providern, die Informationen über den internen systemnahen Aufbau der angeschlossenen Hard- oder Softwarekomponenten enthalten. Durch standardisierte Schnittstellen zu beiden Seiten isoliert der CIM-Server die Clients von den Providern: Die Clients wissen nicht, wie ihre Anfragen abgearbeitet werden, sie haben nicht einmal das Wissen über die Existenz von Providern. Andererseits erhalten die Provider keine Informationen über den Ursprung der an sie gesendeten Befehle. Diese Isolation hat den Vorteil, dass ein CIM-Client, der beispielsweise die CPU-Auslastung eines Servers wissen möchte, die gleiche Anfrage an Unix-, Windows- oder Solarisbasierte Rechner senden kann und jeweils eine Rückantwort im selben Format erhält. Obwohl die tatsächlichen Mechanismen, die diesen Wert ermitteln, sich voneinander unterscheiden, bleibt die Art des Zugriffs also in allen Fällen gleich.
Der wichtigste Teil des *CIM-Servers* ist der *CIM Object Manager (CIMOM)*, der die in einem Repository abgelegten Modellinformationen dazu benutzt, Anforderungen und Rückantworten zwischen den *CIM-Clients*, *-Providern* und *-Listenern* zu koordinieren.

Die *CIM-Clients* interagieren mit dem CIMOM, um auf Anweisung eines Operators das Modell zu modifizieren. Clients können sich sowohl auf der Workstation eines Administrators, als auch auf dem *Managed Node* selbst oder auf einem anderen Gerät befinden.
CIM-Provider arbeiten als Treiber und Schnittstelle zwischen der abstrakten Modellwelt und den hardwarenahen Gegebenheiten der tatsächlichen Hard- und Software, die gemanaged wird. Sie übersetzen die ankommenden Anfragen in tatsächliche Befehle, um den gewünschten Wert zu ermitteln.
CIM-Listener empfangen Informationen über aufgetretene Ereignisse (*Events*) an denjenigen *Managed Nodes*, an denen ein Operator zuvor sein Interesse bekundet hat. [Hob04]

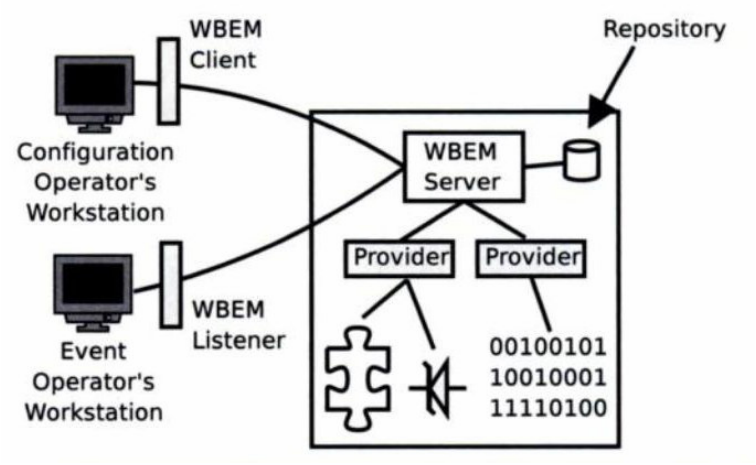

Abbildung 4: Aufbau WBEM-Architektur [Hob04]

Jede Kommunikation zwischen den Management-Clients und der Management-Infrastruktur findet mittels des CIM-XML-Protokolls statt. Es definiert *CIM-Messages*, wohldefinierte Datenpakete, die dazu dienen, CIM-Information auszutauschen. CIM-XML benutzt neben dem CIM auch ein spezielles Mapping von CIM auf XML, das xmlCIM, als Format für die übertragenen Daten. Außerdem existiert einen Satz von Operationen, um CIM-Daten anzufordern und zu manipulieren und schließlich noch eine Kapselung für das HTTP-Protokoll, das als Beförderungsmittel für die *CIM-Messages* dient. [DMTF06a]

2.5.4 Common Information Model (CIM)

Es gibt in heterogenen IT-Infrastrukturumgebungen keine allgemeinen Vereinbarungen über die Informationen, die zu Managementzwecken ermittelt, ausgetauscht und gespeichert werden müssen. Das Informationsmodell spielt daher eine zentrale Rolle innerhalb einer Managementarchitektur. Es spezifiert die Methoden zur Modellierung und Beschreibung von Managementobjekten und legt damit fest, welche Eigenschaften von Ressourcen und welche Vorgänge relevant sind.
Das *Common Information Model (CIM)* bildet deshalb den wichtigsten Bestandteil der WBEM-Architektur. Bei seiner Entwicklung wurde berücksichtigt, dass sich bereits vorhandene Informationsmodelle (wie z.B. SNMP-SMI) möglichst verlustfrei auf CIM abbilden lassen. [Heid03]
Nach der Spezifikation der DMTF ist CIM in drei Schichten unterteilt (siehe Abbildung 5):

- Das *Core Model* ist ein zentraler Kern von einigen wenigen Klassen, Assoziationen und Eigenschaften, die eine Basis für Verfeinerungen und ein grundlegendes Vokabular für die Analyse und Beschreibung zu verwaltender Systeme bilden.

- Das *Common Model* besteht aus einer Menge von Klassen die konkreter gefasst sind als im *Core Model* und bereits als Basis für eine Reihe von Managementanwendungen dienen können. Sie sind aber immer noch unabhängig von bestimmten Technologien

und Implementierungen. Abgedeckt werden hierbei Bereiche wie Systeme, Applikationen, Netze, Endgeräte und Benutzerverwaltung, wobei mit fortschreitender Entwicklung des Modells neue Anwendungsfelder hinzukommen können.

- *Extension Schemas* stellen technologiespezifische Erweiterungen des *Common Model* dar und sind auf spezielle Umgebungen (insbesondere Betriebssysteme) zugeschnitten. (z.B. *Win32 Schema*, WBEM-Implementierung von Microsoft)

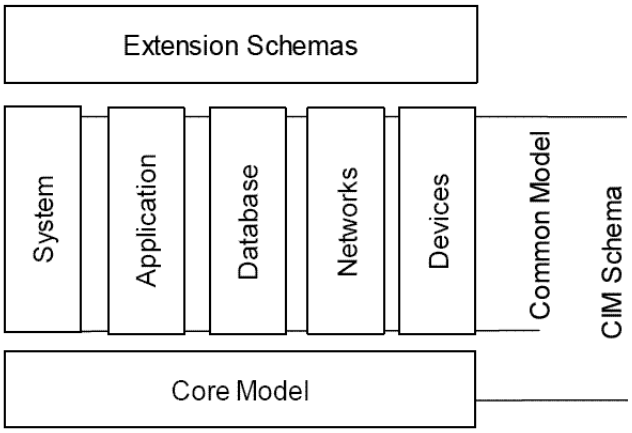

Abbildung 5: Aufbau des Common Information Model (CIM)

Das *Core Model* und das *Common Model* zusammen bilden das *CIM Standard Schema* oder einfacher *CIM Schema*. Es wird von der DMTF ständig erweitert und in seinen verschiedenen Entwicklungsstufen (zur Zeit die Versionen 2.13 bis 2.17) veröffentlicht.
Die Sprache, um CIM-Objekte in textueller Form zu beschreiben nennt sich *Managed Object Format (MOF)* und basiert auf der *Interface Definition Language (IDL)*. [DMTF06a]

3 ANALYSE VON STEINMAYR NSM ANHAND EINES BEISPIELSZENARIOS

In diesem Kapitel wird eine Einführung in die Software *Netz-Service-Management* (NSM) der Fa. Steinmayr Net Intelligence GmbH gegeben. Steinmayr NSM ist ein über diverse Versionen gewachsenes Programm.
NSM dient vornehmlich der Inventarisierung und Katalogisierung von Telekommunikations- (TK) und IT-Infrastruktur. Darüber hinaus werden Verknüpfungen zwischen technischen Komponenten erfasst. Weitere Funktionen gibt es für die Planung von neuer Infrastruktur.
Die Grundlegende Funktionalität von NSM kann im *Asset-Management* angesiedelt werden, Teile gehen aber darüber hinaus. Hier übernimmt die Software Aufgaben, die charakteristisch für eine CMDB sind. NSM lässt sich daher nicht klar in eine der Managementdomänen einordnen.
In diesem Kapitel werden die Kernbestandteile *Cable* und *Logic* von NSM erläutert und einige grundlegende Arbeitsabläufe dokumentiert. NSM baut auf einer eigenen Begriffswelt auf, in der sich Informationen über die Infrastruktur und deren Komponenten fassen lassen; diese ist aber proprietär und wird von anderen *Asset-Management*-Produkten am Markt nicht genutzt. Für ein grundlegendes Verständnis des Leistungsumfangs von NSM ist es daher notwendig, einen Überblick über die verwendeten Begrifflichkeiten zu haben. Dieser wird in diesem Kapitel gegeben.
Es lassen sich Informationen wie Position, Konfiguration, Dokumentation für Netze unterschiedlicher Größe zentral vorhalten und bei Bedarf aktualisieren. *Asset-Management* stellt ursprünglich den Kern der Software NSM dar.

IT-Komponenten in Infrastrukturen gewinnen entscheidend an Bedeutung gegenüber Komponenten aus dem TK-Bereich. Damit sind auch die Anforderungen an eine Software wie NSM gestiegen. Hinzukommende Komponenten gemeinsam mit den bereits existierenden an einer zentralen Stelle verwalten können. Gleichzeitig sind IT-Komponenten flexibel einsetz- und anpassbar; ihre Einsatzmöglichkeiten umfassen neue Anwendungsfelder. Für NSM ergeben sich damit steigende Anforderungen an die Funktionalität. Das Datenmodell muss hinreichend flexibel sein, sehr unterschiedliche Komponenten der IT- und TK-Welt aufnehmen zu können, darf aber nicht inkompatibel zu den bestehenden Datenbeständen in den Datenbanken der Kunden werden.
Um diesen Anforderungen an das IT-Management gerecht zu werden, enthält NSM über das *Asset-Management* hinaus Bestandteile, die weit über das *Asset-Management* hinaus gehend andere Bereiche des IT-Managements wie das *Configuration Management* hineinreichen.
Für ein grundlegendes Verständnis des Leistungsumfangs von NSM ist es daher notwendig, einen Überblick über die verwendeten Begriffe zu haben. Dieser wird in diesem Kapitel gegeben.
Anhand eines Beispielszenarios, welches sich ansatzweise an einem Teil der Infrastruktur der ATIS orientiert, wird eine Testkonfiguration der Software dargestellt. Darauf aufbauend werden einige erweiterte Funktionen von NSM beschrieben.
Für die Analyse von NSM in dieser Studienarbeit ist keine formale Beschreibung des Datenmodells von NSM, etwa in Form von UML-Diagrammen, vorhanden. Um trotzdem eine Grundlage für die Arbeit am Datenmodell von NSM zu erhalten, werden hier selbst erstellte Klassendiagramme gezeigt. Diese wurden gewonnen aus den Schema-Daten, die das Datenbank-Administrationsprogramm lieferte und aus der Arbeit mit NSM selbst. Es ist also nur eine Interpretation der gewonnenen Informationen und wurde nicht durch die Fa. Steinmayr als korrekt bestätigt.

3.1 Aufbau und grundlegende Funktionalität von NSM

Um eine Grundlage für die Arbeit mit NSM zu schaffen, wird zunächst die Systemumgebung, in der NSM genutzt werden kann, sowie der Aufbau der hier genutzten Komponenten von NSM beschrieben.

3.1.1 Anforderungen an die Systemumgebung von NSM

Im Rahmen dieses Kapitels wird die Version 6 von NSM betrachtet. NSM ist eine rein Microsoft Windows-basierte Software. Sie stellt keine besonderen Anforderungen an die verwendete Windows-Version bzw. das Setup. Das Windows-basierte Frontend ist zurzeit die einzige Möglichkeit des Zugriffs auf NSM.
Sämtliche in NSM erfassten Daten werden in einer Datenbank abgelegt. Es wird als Datenbank ausschließlich Oracle unterstützt. Diese muss vor der Installation von NSM eingerichtet werden. Es ist darüber hinaus Aufgabe desjenigen, der die Software installiert, das für NSM notwendige Datenbankschema zu importieren.

3.1.2 Aufbau

NSM ist ein Softwaresystem, bestehend aus mehreren Modulen, für das IT-Management. Diese sind Cable-NSM, Logic-NSM und Task-NSM (Abbildung 6).
Cable-NSM wird benutzt für die Erfassung und Dokumentation von Kommunikations- und Datennetzen. Es dient der Planung, Darstellung und Verwaltung der Infrastrukturkomponenten und bildet den Kern von NSM. Sein Datenbestand ist zentraler Teil einer NSM-Installation.
Logic-NSM ist ein weiterer Teil von NSM. Es dient dazu, die in Cable-NSM erfassten aktiven Komponenten mit ihrem realen Gegenstück verknüpfen zu können. Es ist eng an Cable-NSM gekoppelt und erweitert den von Cable-NSM erfassten Datenbestand um eigene Bestandteile. Logic-NSM kommt mit einem von der Hauptapplikation entkoppeltem so genanntem Scan-Server. Dieser läuft unabhängig von Cable- und Logic-NSM, wird aber ausschließlich durch Logic-NSM gesteuert.
Task-NSM ist ein weiterer Bestandteil von NSM. Es ist für Auftragsverwaltung, Helpdesk-Unterstützung und Wissensdatenbanken zuständig. Da dieser Bestandteil für den in dieser Studienarbeit betrachten Teil des IT-Managements nicht relevant ist, wird es hier nicht weiter betrachtet.

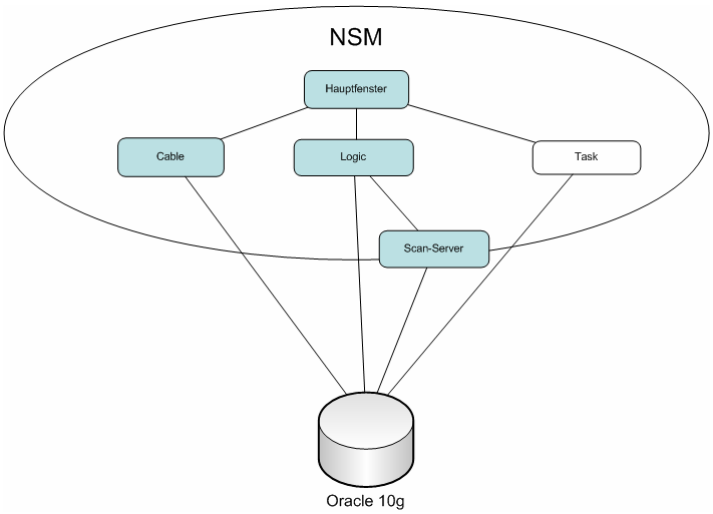

Abbildung 6: Aufbau NSM

NSM wird in unterschiedlichen Konfigurationen ausgeliefert, die jeweils einen an den Bedürfnissen der Kunden ausgerichteten Funktionsumfang besitzen. Module wie das Task-NSM können so weggelassen werden, ohne den Rest der Applikation zu beeinträchtigen.

3.1.3 Ausführung

Auch wenn die Module von NSM als separate Bestandteile erscheinen, werden sie immer über das NSM-Hauptmenü (Abbildung 7) gestartet.

Abbildung 7: NSM-Hauptmenü

Jeder Karteireiter repräsentiert einen Aspekt der NSM-Installation. Sind einzelne Komponenten nicht installiert, erscheinen deren Karteireiter nicht.
NSM enthält eine eigene Nutzerverwaltung, über die der Zugriff auf das Programm geregelt wird. Ein Nutzer muss sich beim Start anmelden und erhält den für ihn vom Administrator konfigurierten Zugriff auf die Programmbestandteile.

3.2 Beispielszenario ATIS

Für die Analyse von NSM als Teil eines integrierten Managements werden im Folgenden die relevanten Komponenten Cable-NSM und Logic-NSM beschrieben. Um gleichzeitig einen Einblick in den Ablauf einer Erfassung einer Infrastruktur mit NSM zu bekommen, werden anhand eines Beispielszenarios die wichtigsten Schritte dokumentiert.
Das hier beschriebene Szenario orientiert sich ansatzweise an der bestehenden Infrastruktur der ATIS. Es werden exemplarisch einige Komponenten in NSM umgesetzt. Anhand einiger existierender aktiver Komponenten im ATIS-Netz wird die Kopplung zwischen Logic-NSM und realer Infrastruktur gezeigt.

3.3 Cable-NSM

Cable-NSM ist der zentrale Bestandteil von NSM. Es dient der Erfassung und Planung von Infrastrukturkomponenten, Verbindungen und Standortinformationen.
Die Netzverwaltung mit Cable-NSM gliedert sich in zwei Bereiche, *Stammdaten* und *Definitionen*. *Stammdaten* enthalten eine möglichst vollständige Datenbank aller Komponenten der Infrastruktur incl. Konfigurationsdaten wie Rufnummern bei TK-Komponenten oder IP-Adressen von IT-Komponenten. Um festzulegen, welche Komponenten in dieser Datenbank erfasst werden können, werden diese unter *Definitionen* abgelegt. In dieser Bibliothek werden damit in alle in der Infrastruktur genutzten Komponenten katalogisiert.

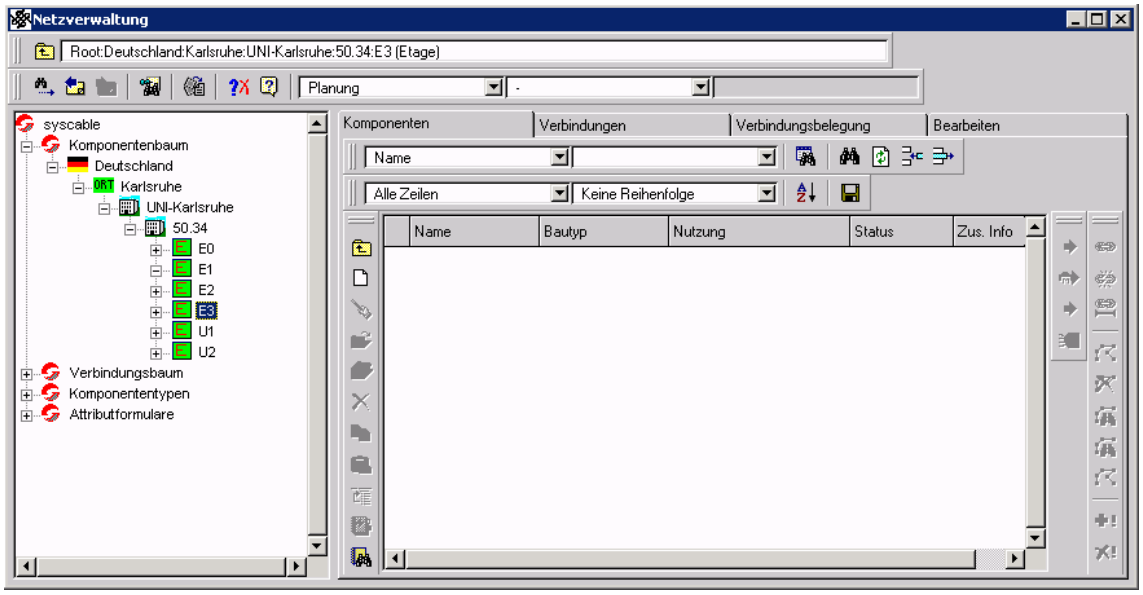

Abbildung 8: Cable-NSM

3.3.1 Standardansicht Cable-NSM

Die Ansicht *Netzverwaltung* (Abbildung 8) zeigt auf der linken Seite eine hierarchische Darstellung mehrerer Baumstrukturen. Diese werden unter einem Wurzelknoten *syscable* zusammengefasst.
Zentraler Teil von Cable-NSM ist der Komponentenbaum. Ausgeklappt enthält dieser sämtliche erfasste Komponenten. Die Hierarchie orientiert sich an der geographischen Position des jeweiligen Teils. Daher werden auch Standortinformationen wie Land, Stadt, Gebäude usw. erfasst, bis hin zur eigentlichen Komponente.
Auf der rechten Seite werden Informationen im Kontext der jeweils ausgewählten Komponente angezeigt. Je nach Komponententyp werden Ansichten als Karteireiter ein- oder ausgeblendet. Dies sind untergeordnete Komponenten (immer sichtbar), grafische Darstellung (z.B. für Schränke), Verbindungen o.ä.

3.3.2 Erfassung von Komponenten

Komponenten umfassen alle aktiven und passiven Elemente der Infrastruktur. Dazu gehören statische Elemente der Umgebung wie Gebäude, Etagen, Räume und sämtliche Bestandteile der IT-Infrastruktur wie Dosen, Patchfelder, Server, Switche, PCs usw.
Komponenten werden eingefügt in einen Komponentenbaum (Abbildung 9). Dieser gibt in einer hierarchischen Struktur den genauen Standort und die Zugehörigkeit jeder Komponente wieder, indem jedem Knoten ein Vaterknoten zugeordnet ist. Alle Komponenten haben einen Bautyp (siehe Komponententyp) und weitere zugeordnete Informationen wie Namen und einen Status, mit dem sie z.B. als defekt oder inaktiv markiert werden können.
Zur besseren Übersicht sind Komponenten mit grafischen Darstellungen versehen; außerdem können diese genutzt werden, um beispielsweise die Verkabelung in einem Kabelschrank zu erleichtern, indem die geplante Verkabelung vorher bereits schematisch in Cable-NSM umgesetzt wird.

3.3.3 Attributfelder

Komponenten können frei definierbare Attribute zugeordnet werden, deren Art und Einsatz dem Anwender überlassen werden. Hierdurch kann der Anwender in gewissem Maße Einfluss auf die Erfassung der Daten in NSM nehmen; eigene Formulare können integriert werden, deren Eingabefelder als besagte Attribute in der Datenbank abgebildet werden.

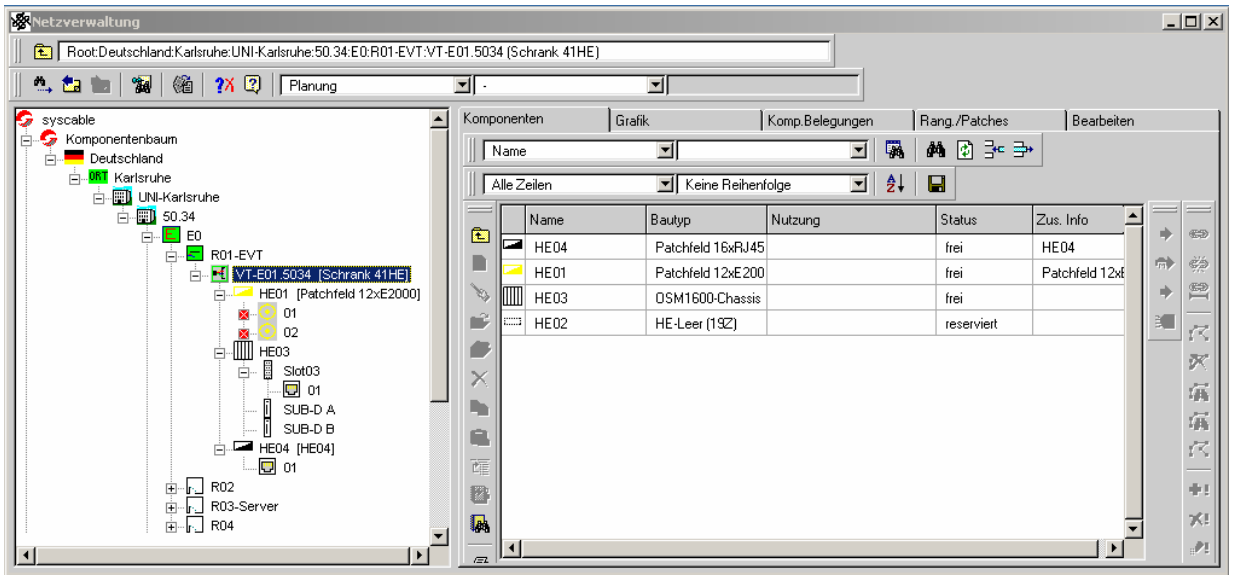

Abbildung 9: Erfassung eines Kabelschrankes im Komponentenbaum

3.3.4 Erfassung von Komponententypen

Sämtliche zu erfassenden technischen Komponenten sind jeweils einem Komponententyp zugeordnet. Cable-NSM wird ausgeliefert mit einer Basisbibliothek von Komponententypen. Die Erfassung weiterer Komponententypen bleibt entweder dem Benutzer überlassen oder kann von der Firma Steinmayr übernommen werden. Kunden können zur schnelleren Erfassung ihrer Infrastruktur über das Internet auf einen umfangreichen Datenbestand von existierenden Komponententypen zugreifen; dieser wird von Steinmayr gepflegt und bei Bedarf um neue Komponenten erweitert. Daher erfordern neue Komponenten in der Infrastruktur eine ständige manuelle Anpassung des Datenbestandes der Komponententypen. Die Pflege der Daten erfolgt völlig unabhängig vom Hersteller der Komponenten ausschließlich durch Steinmayr bzw. den Benutzer.
Der Screenshot in Abbildung 10 zeigt ein Fenster von Cable-NSM mit einigen Komponententypen.

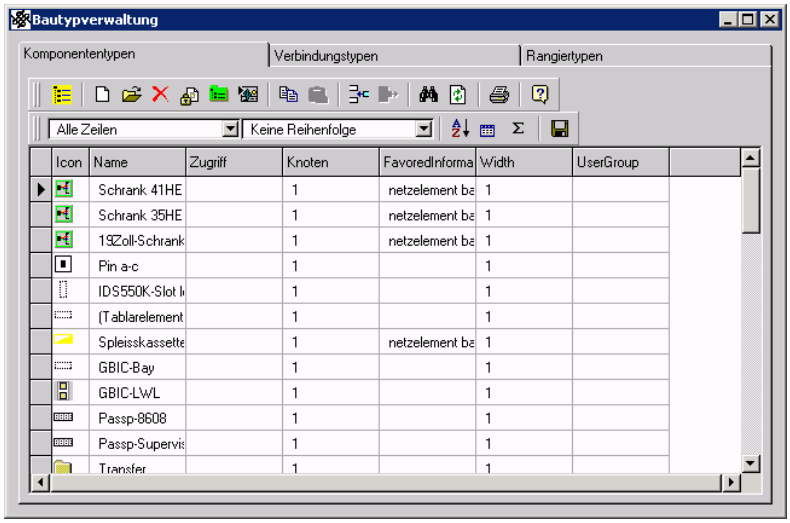

Abbildung 10: Verwaltung von Komponententypen

3.3.5 Nutzung

Komponenten kann eine Eigenschaft *Nutzung* (Abbildung 11) zugeordnet werden. Dazu gehören u.a. eine Nutzungsart, ein Nutzungsname und ein Status.
Die Nutzung gibt im Rahmen von NSM einer Leitung eine Bezeichnung. Bei IP-Netzen ist die Nutzungsart *IP*; unter *Nutzungsname* wird die IP-Adresse einer Komponente erfasst.
Mit der Nutzung wird so die Adressierung von Komponenten dokumentiert.

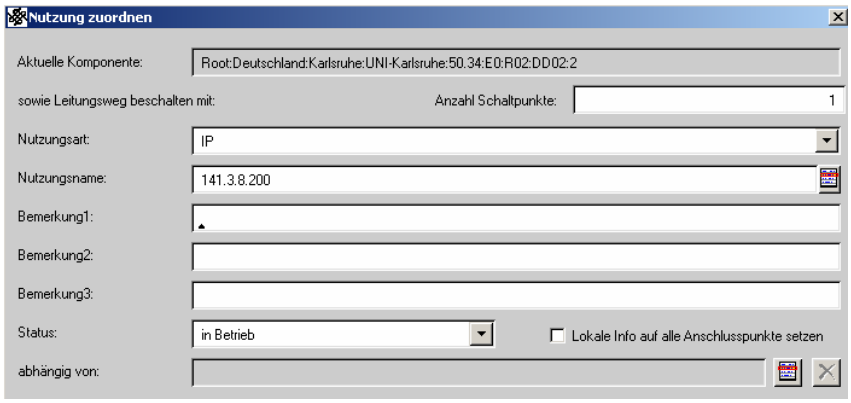

Abbildung 11: Eine Nutzung zuordnen

3.3.6 Erfassung von Verbindungen

Zwischen den im Komponentenbaum erfassten Anschlüssen können Verbindungen erfasst werden. Wie Komponenten sind auch Verbindungen einem Bautyp zugeordnet.
Verbindungen werden in NSM in einer weiteren Baumstruktur dargestellt, dem Verbindungsbaum (Abbildung 12). Eine Verbindung wird erfasst, indem zu verbindende Komponenten im Komponentenbaum angelegt werden, eine Verbindung im Verbindungsbaum angelegt wird, und über *Kabel auflegen* die Verbindung zwei Anschlüssen an Komponenten zugeordnet wird.

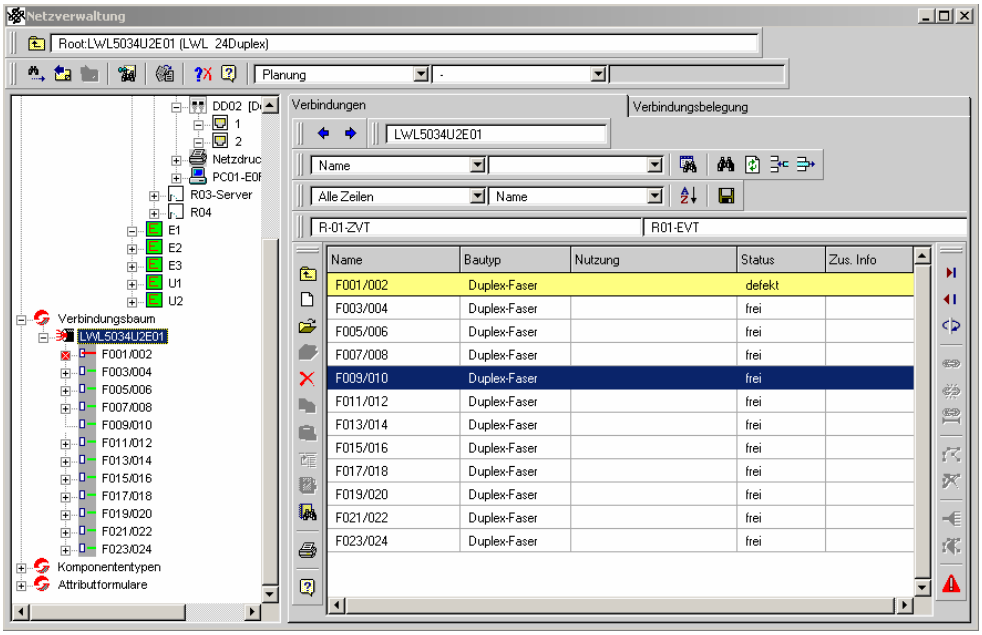

Abbildung 12: Cable-NSM Verbindungsbaum

3.3.7 Umsetzung des Beispielszenarios in Cable-NSM

Als Grundlage eines beispielhaften Setups für NSM wird ansatzweise ein kleiner Teil der Infrastruktur der ATIS in Cable-NSM umgesetzt.
Zunächst wird das Informatik-Gebäude der Universität Karlsruhe erfasst. Dafür wird der Standort erfasst, die Universität, und das Gebäude. Im Sinne von NSM sind dies auch Komponenten bestimmter Bautypen wie *Ort* und *Standort*. In dem Gebäude *50.34* werden mehrere Stockwerke erfasst, *E3-E0* und *U1-U2*.
In dem Stockwerk *E0* wird ein Etagenverteilraum *R01-EVT* eingefügt. In ihm wird ein Schrank angelegt mit mehreren Höheneinheiten und einem Patchfeld mit zwei LWL-Anschlüssen. Ein zweiter Raum wird angelegt, *R02*. In ihm werden 2 Datendosen mit RJ-45-Anschlüssen angelegt, ein Netzwerkdrucker und ein PC. Ein dritter Raum *R03-Server* enthält einen Switch *ATIS-Switch*, einen weiteren Netzwerkdrucker und drei Server.
Im Stockwerk *U2* befindet sich die Komponenten der zentralen Infrastruktur.
Dieses Setup spiegelt einen kleinen Teil der Struktur der ATIS wieder. In jeder Etage befindet sich ein Verteilerraum, in dem Verbindungen aus der zentralen Infrastruktur ankommen und von dort aus, für die Verteilung aufbereitet, Leitungen in die Etagenräume laufen.
Die erfassten Daten in Cable-NSM ließen sich entsprechend erweitern, um den realen Gegebenheiten zu entsprechen. Dies ist für die Analyse im Rahmen dieser Studienarbeit aber nicht notwendig.

3.3.8 Datenmodell von Cable-NSM

Die in Cable-NSM abgelegten Daten bilden den Kern einer NSM-Installation. Sie werden in Tabellen einer Oracle-Datenbank abgelegt. Das Datenmodell wird dafür in ein relationales Modell abgebildet.
Namen und Assoziationen der Tabellen der Datenbank orientieren sich eng an den in 3.3.2 und 3.3.4 beschriebenen Begriffen und grundlegenden Funktionen der Anwendung (Abbildung 13).

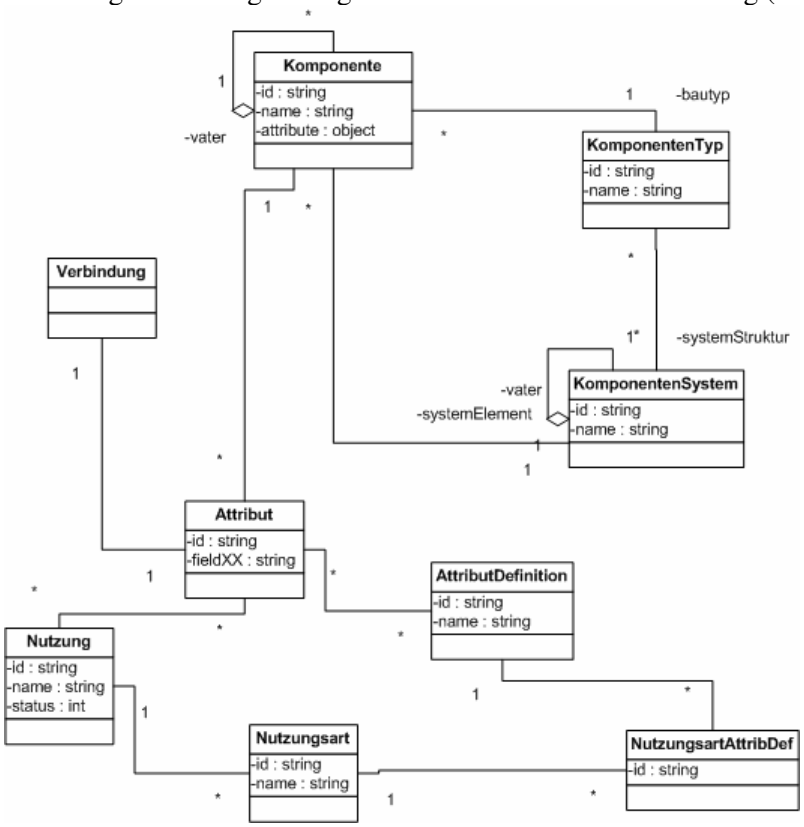

Abbildung 13: Auszug Klassendiagramm für das Datenmodell von Cable-NSM

3.4 Logic-NSM

Logic-NSM ist ein Softwaremodul und Bestandteil von NSM. Es dient der Erfassung von aktiven Infrastrukturkomponenten und dem Abgleich der in Cable-NSM erfassten aktiven Komponenten mit den im Netz erreichbaren. Der Fokus von Logic-NSM liegt damit auf *Audits* der Infrastruktur. Die Datenerfassung selbst wird dabei über einen Scan- und Importserver abgewickelt. Dieser läuft separat von NSM und führt Aufträge durch, wenn diese von NSM erteilt werden.
Um aktive Komponenten in einem Netz zu ermitteln, wird eine Subnetz-Suche erzeugt. Ein zusätzlich erstellter Scan-Job spezifiziert Zeitpunkt und eventuelle Wiederholungen der Suche und aktiviert den Scan-Server. Dieser führt den Scan-Job durch und legt die gefundenen Objekte, sogenannte ClientScanner, in einer Tabelle ab. Erfasst werden IP-Adresse und, wenn möglich, MAC-Adresse und Hostname. Durch die Asset-Zuordnung im Logic-NSM können diese dann den entsprechenden Komponenten zugeordnet werden (AssetMapper). Die Zuordnung wird gespeichert und bleibt so für zukünftige Suchläufe erhalten.

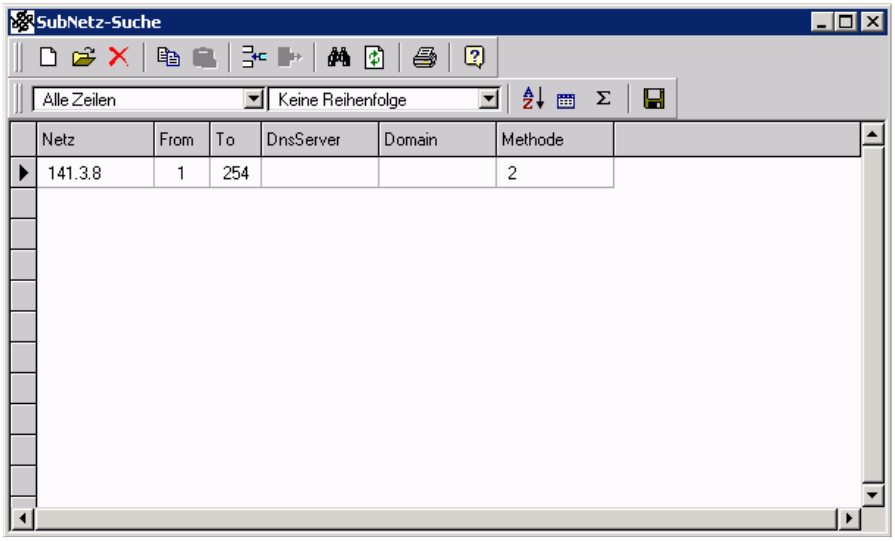

Abbildung 14: Objektsuche mit Logic-NSM

Logic-NSM unterstützt mehrere Protokolle, über die es Daten mit aktiven Komponenten austauschen kann:

- Von Windows-Computern können Informationen über die WMI-Schnittstelle abgefragt werden; entsprechend konfigurierte Windows-Rechner sind so beliebig detailliert erfassbar. Darüber hinaus besteht bei Windows-Computern die Möglichkeit der Abfrage über die Registry.
- Per SNMP können Informationen über aktive Netzwerkkomponenten, die diesen Standard unterstützen, erfasst werden

Durch den Scan-Server gewonnene Daten werden als Attribute der entsprechenden Komponenten in der Datenbank abgelegt. Dazu wird eine Scan-Konfiguration angelegt und mit einer Abfrage (Query) und abzufragenden Komponenten verknüpft. Um die Abfrage durchzuführen, wird ein Scan-Job für den Scan-Server angelegt. Dieser kann zu bestimmten Zeitpunkten ausgeführt und automatisch wiederholt werden.

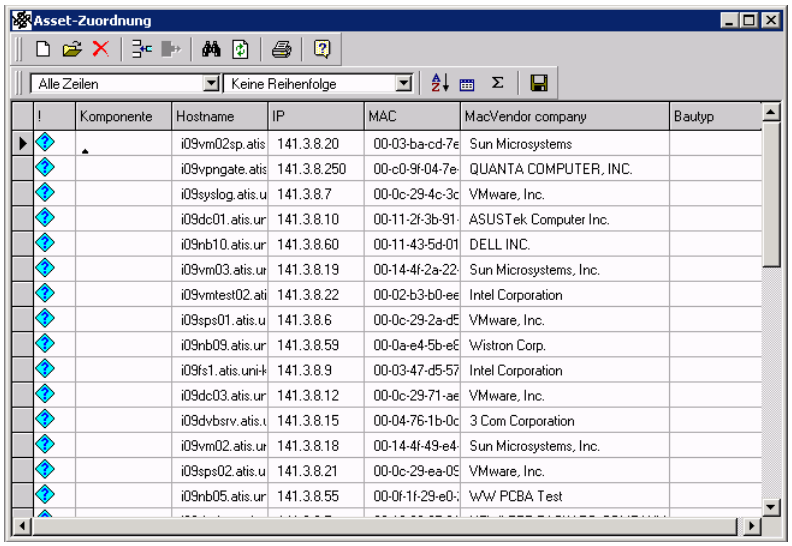

Abbildung 15: Asset-Zuordnung mit Logic-NSM

3.4.1 Software–Erfassung in Logic-NSM

Über die WMI-Management-Schnittstelle können von Windows-Computern detaillierte Informationen über die auf ihnen installierte Software erfasst werden. Diese Funktionalität kann von einer *Asset-Management*-Software wie NSM genutzt werden, um einen automatisch aktualisierbaren Datenbestand über die auf Windows-Computern installierte Software zu erhalten.
Logic-NSM kann diese Schnittstelle nutzen und die durch sie abgerufenen Informationen entsprechend zuordnen. Hierfür wird ein *ClientScanner* einem Windows-Rechner zugeordnet. Bei einem Scan werden die über die WMI-Schnittstelle abgerufenen Informationen als Attribute der entsprechenden Komponente in der NSM-Datenbank abgelegt.
Diese Informationen eignen sich, um im Rahmen des *Asset-Managements Audits* der auf am Netz angeschlossenen Rechnern installierten Software durchzuführen. Dies kann zur Bestandsanalyse hilfreich sein.
Nicht möglich ist die Erfassung von Software als eigenständige Objekte im Komponentenbaum. Daher können elementare Eigenschaften wie Versionsnummern, separat aufgeführter Produktname und –hersteller usw. nicht als eigenständige Attribute erfasst werden, was den Nutzen dieser Informationen im Rahmen von technischer Nutzung für andere Management-Prozesse stark beschränkt.

3.4.2 Umsetzung des Beispielszenarios in Logic-NSM

Um die Funktionalität von Logic-NSM in der ATIS demonstrieren zu können, ist die NSM-Installation in das Mitarbeiternetz integriert. So können beispielhaft einige Netzwerkgeräte erfasst werden.
Es wird das WMI-Protokoll zur Abfrage der Informationen genutzt. Daher können bei dieser Suche ausschließlich Informationen über Windows-Rechner berücksichtigt werden.
Um die Erfassung durchzuführen, wird

- Der Scanserver installiert; er erhält Zugriff auf den IP-Bereich Mitarbeiternetzes
- Eine Subnetz-Suche angelegt für einen IP-Adressbereich des ATIS-Mitarbeiternetzes
- Eine Scan-Job vom Typ *ClientScanner* angelegt und für die Suche das erstellte Subnetz ausgewählt
- Für diesen Scan-Job ein Startzeitpunkt festgelegt

Der Scan-Server greift direkt auf die Datenbank von Logic-NSM zu. Er erkennt, wann entsprechende Scan-Jobs auszuführen sind, führt diese aus, und schreibt die Ergebnisse in die Datenbank zurück.
Nach dem Scan können die Ergebnisse in Logic-NSM betrachtet werden. Ein Scan-Protokoll gibt Auskunft über Zeitpunkt und Verlauf der Scans.

Die beim Scan gefundenen Objekte können Komponenten im Komponentenbaum zugeordnet werden. Dies dient der Erfassung von Attributen von Komponenten, beispielsweise installierter Software bei Windows-Rechnern, und der einfacheren Aktualisierung des Datenbestands von Cable-NSM. Hierfür wird in der Asset-Zuordnung über *neu* parallel zur Objekt-Liste der Komponentenbaum eingeblendet. Über eine Auswahl in beiden Ansichten können Zuordnungen durchgeführt werden (Abbildung 16).

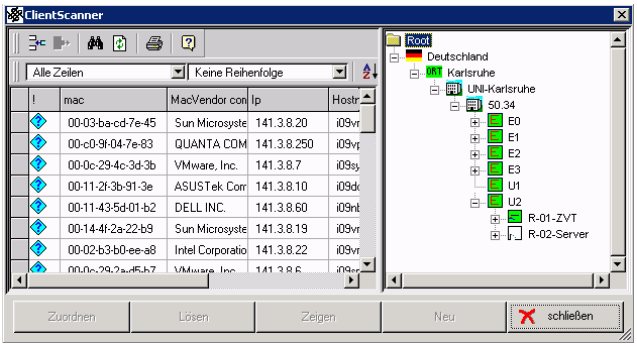

Abbildung 16: Clientscanner-Zuordnung

3.4.3 Datenmodell von Logic-NSM

Wie für Cable-NSM gilt auch für Logic-NSM, dass sämtliche Daten in einer Oracle-Datenbank abgelegt werden.
Das in Abbildung 17 zu sehende Klassendiagramm bildet einen kleinen Teil des Datenbankschemas von Logic-NSM ab.

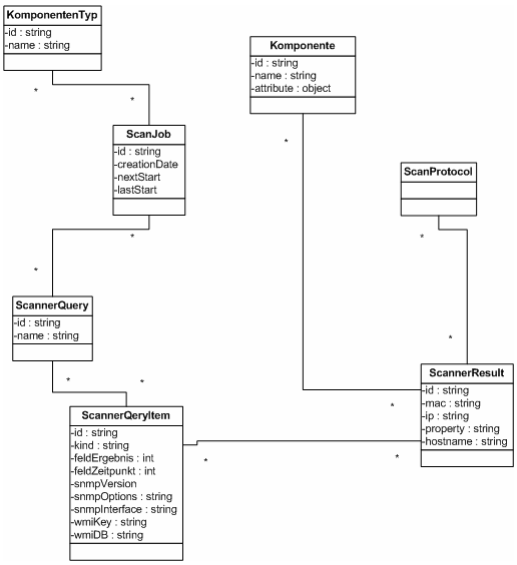

Abbildung 17: Auszug Klassendiagramm für das Datenmodell von Logic-NSM

4 NSM IM RAHMEN EINES SERVICEORIENTIERTEN IT-MANAGEMENTS

NSM wird in diesem Kapitel im Kontext des IT-Managements betrachtet und ein Konzept entwickelt, wie es in einer Architektur für serviceorientiertes IT-Management genutzt werden kann. Es wird daher in diesem Kapitel untersucht, an welchen Stellen Anknüpfungspunkte vom *Asset-/Configuration-Management*, wie es durch NSM realisiert ist, hin zu serviceorientiertem IT-Management bestehen.
Es wird, ausgehend von einer Dienstbeschreibung durch *Service Specification Sheets*, dargestellt, ob sich die in einer NSM-Installation enthaltenen Informationen mit Diensteigenschaften in Verbindung gebracht werden können. So soll ein Ansatzpunkt gefunden werden, wie NSM Teil einer serviceorientierten Managment-Architektur werden kann.
Das NSM-Datenmodell bzw. dessen zentrale Bestandteile Cable- und Logic-NSM werden auf eine Erweiterung hin zu einer serviceorientierten Sicht auf Infrastrukturkomponenten untersucht. Im Mittelpunkt der Erweiterung steht die Einführung eines *Service Specification Sheets* (SSS) als Werkzeug des IT-Managements.

4.1 NSM und IT-Management

Mit steigender Bedeutung von IT-Services als Produkt von Infrastruktur-Providern steigt auch deren Bedeutung für das IT-Management. Allerdings verlangt ein am Service orientiertes IT-Management eine ganzheitliche Sichtweise auf alle Teile der Infrastruktur. Mit dem Service als Endprodukt werden automatisch auch die für das Management genutzten Software und Prozesse in dessen Kontext gestellt.
Dies gilt also auch für Software aus dem *Asset-/Configuration-Management* wie NSM. Der hier beschriebene Ansatz erweitert NSM um ein Modell, das Aspekte eines IT-Services umsetzt. Dabei wird darauf geachtet, dass sich eine solche Erweiterung umsetzbar bleibt, ohne an der Kernfunktionalität an sich Änderungen zu erfordern. Es soll ein problemloser Übergang von der bisherigen Nutzung hin zu einer erweiterten, serviceorientierten Nutzung möglich sein. Insbesondere sollen Veränderungen am Datenmodell von NSM möglichst wenig umfangreich ausfallen.

4.1.1 Das Service-Objekt als Kapselung eines IT-Services

Ein Service-Objekt beschreibt einen IT-Service als Leistung des *Service-Providers* für den Service-Kunden. Es dient als Darstellung des IT-Services für den *Service-Provider* und orientiert sich an einem *Service Specification Sheet*.

4.1.2 NSM und IT-Services

Die Sicht von NSM auf eine Infrastruktur ist rein lokationsorientiert. Zu jeder Infrastrukturkomponente ist ersichtlich, an welchem Standort sie sich befindet, und welche technischen Verbindungen zwischen ihr und anderen Komponenten bestehen.
Diese Information ist oft nicht hinreichend.

- Es ist allein vom Standort von Komponenten ausgehend nicht ersichtlich, inwiefern welche Komponente zu welchem IT-Service beiträgt.
- Es ist nicht ersichtlich, welche Software, die auf einer der Komponenten läuft, zu welchem Dienst beiträgt
- Es gibt keinerlei Möglichkeit, Zuständigkeiten für die Administration der Infrastruktur zu erfassen
- Anforderungen an Eigenschaften der verwendeten Komponenten, die sich aus zu erbringenden IT-Services ergeben, werden nicht erfasst

4.1.3 Anforderungen an NSM für das IT-Service-Management

Daher wird in dieser Arbeit eine Erweiterung der Funktionalität von NSM vorgeschlagen. Diese muss sich an den durch NSM gegebenen Rahmenbedingungen orientieren:

- Eine Integration in die Benutzerschnittstelle von Cable-NSM soll problemlos möglich sein
- Die vorgeschlagenen Erweiterungen des Datenmodells sollen möglichst klein bleiben, um den Implementierungsaufwand niedrig zu halten

4.1.4 Service Specification Sheets als Grundlage einer Service-Sicht in NSM

Eine Anwendung wie NSM mit einer serviceorientierten Sicht zu verknüpfen wirft die Frage auf, inwieweit dieses praktikabel ist. Es ist zu klären, wie der Begriff *Serviceorientierung* im Kontext einer Software wie NSM zu interpretieren ist.
Es gibt bis jetzt keinen Best-Practice-Ansatz, für die Betrachtung von Serviceorientierung und *Asset-/Configuration-Management* im Kontext einer Software wie NSM. In diesem Kapitel wird deshalb das *Service Specification Sheet* (SSS) als Grundlage eines Service-Begriffs für NSM genutzt. Es wird, ausgehend vom *Asset-Management*, ein praktikabler Ansatz beschrieben, wie eine Weiterentwicklung von NSM möglich wäre, ohne allzu große Änderungen am bestehenden Datenmodell und damit an fertigen Installationen zu verlangen.

4.1.5 Service Specification Sheets als Werkzeug eines Service-Providers

Es gibt kein formalisiertes Verfahren, wie ein SSS für einen IT-Service erstellt wird, und keine feste Struktur, wie es auszusehen hat. Ein *Service Specification Sheet* kann im einfachsten Fall ein reines Text- oder Tabellendokument sein, das als Dokumentation für den *Service Manager* über bestehende Zuständigkeiten und Abhängigkeiten dient.
Viele Informationen über eine Infrastruktur werden bereits erfasst, wenn ein Provider Tools für das *Asset-Management* einsetzt. Diese enthalten bereits viele Informationen, die auch in einem SSS eine wichtige Rolle spielen. Ließen sich die in einer Asset-Datenbank enthaltenen Daten für ein SSS nutzen, wäre dies ein Schritt in Richtung eines integrierten Service-Managements. Es entstünde eine Verbindung von einer reinen Asset-Datenbank, wie sie bei vielen Infrastrukturbetreibern bereits existiert, hin in Richtung einer an SLAs orientierten Sichtweise.

4.1.6 Service Specification Sheets als Teil einer CMDB

NSM hat ein proprietäres Datenmodell. Cable-NSM und Logic-NSM sind auf die Erfassung rein technischer Aspekte einer Infrastruktur ausgerichtet. Das steht einer Service-Beschreibung in einer integrierten CMDB entgegen, da sich die Datenmodelle der Applikationen stark unterscheiden und, wenn überhaupt, nur mit beträchtlichem Aufwand Verknüpfungen von einer IT-Management-Applikation zur anderen herstellen lassen.

4.2 NSM und Serviceorientiertes IT-Management

NSM soll um eine weitere Komponente erweitert werden, das Service-NSM. Dieses implementiert ein *Service Specification Sheet* so, dass es sich gut in die Datenbasis existierender Installationen integrieren lässt.
Es wird ein praktischer Ansatz verfolgt, wie für die Umsetzung eines SSS die bestehende Datenbasis von NSM genutzt werden kann; das SSS soll sich in NSM integrieren und dem Nutzer einen Mehrwert an Information bieten.
Damit wird in NSM umgesetzt, was im Rahmen des IT-Managements weiter an Bedeutung gewinnt; es wird für *Service-Provider* möglich, die von ihnen bereitgestellten IT-Services in einer CMDB zu erfassen, ohne dass sich die Komplexität der CMDB nennenswert erhöht.

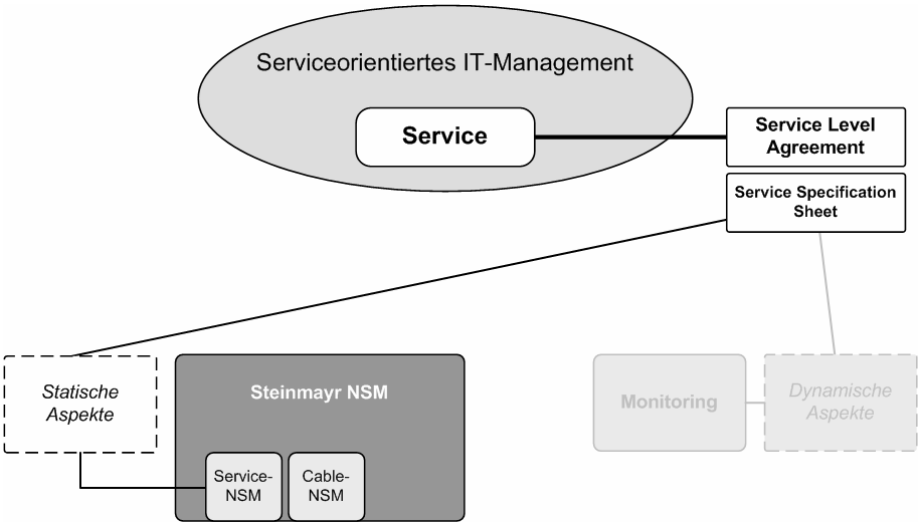

Abbildung 18: Service-NSM als Teil eines serviceorientierten IT-Managements

4.3 Konzept zur Modellierung einer Service Specification in NSM

Kern von NSM ist die Darstellung des Komponentenbaums in Cable-NSM. wird ein *Servicebaum* eingeführt. In dem im Folgenden beschriebenen Konzept für die Modellierung eines SSS wird ein weiteres strukturierendes Element *Servicebaum* analog zur Komponentenbaum von Cable-NSM eingeführt. Im Mittelpunkt steht dabei ein neu eingeführtes *Service*-Objekt. Es bildet die Grundlage für alle im Service-Baum erfassten Knoten. Eines (ohne Vater-Knoten) bildet das Wurzelobjekt. Seine Funktion ist, dass es als Vater für die Servicehierarchie dient. Es ist, wie die Komponente im Komponenten-Baum von Cable-NSM, generisch ausgelegt.
An jedes Service-Objekt können weitere Service-Objekte angehängt werden. Durch die im Service-Baum an das Service-Objekt angehängten weiteren Objekte entsteht eine Darstellung der in diesem Kontext relevanten Informationen über den IT-Service.
Der Servicebaum stellt eine Modellierung des IT-Service dar; die Idee ist, dass nicht nur ein Service mit beliebigen Attributen erfasst werden kann, sondern

- Abhängigkeiten von Ressourcen, diese können sein
  - Andere, auch in NSM erfasste, interne IT-Services
  - Externe IT-Services
  - Komponenten, die in Cable-NSM erfasst sind
  - Operation Level Agreements
- OLAs, in denen Zusicherungen für die Qualität von erbrachten Leistungen der Infrastruktur gegeben werden

Mit diesem Ansatz soll die Modellierung eines IT-Service im Sinne eines SSS ermöglicht werden. Es sollen sich beliebige IT-Services erfassen lassen, und das dem Service-Baum zugrunde legende Datenmodell muss hinreichend flexibel sein, um unterschiedlichen Organisationsformen von Unternehmen gerecht zu werden.

4.3.1 Abhängigkeiten von anderen Services

Es ist sinnvoll, die Ressourcen einer Infrastruktur in Ebenen einzuteilen:

- Unter *Anwendungen* werden alle Softwarekomponenten erfasst
- *System* umfasst Hardware, auf der die unter *Anwendungen* erfassten Softwarekomponenten betrieben werden
- *Netz* umfasst die zur Kommunikation der Systeme notwendigen aktiven und passiven Netzwerkkomponenten

Diese Unterteilung macht auch im Kontext einer Service-Darstellung Sinn. Sie erhöht die Übersichtlichkeit, da sich anhand der Darstellung eines Services leicht erkennen lässt, inwiefern dieser auf Komponenten unterschiedlicher Ebenen angewiesen ist.
Eine Möglichkeit besteht darin, jede Komponente, für die eine Abhängigkeit von einem Service besteht, in jedem Service-Kontext neu zu erfassen.
In einem solchen Service-Baum entstehen leicht Redundanzen: Netz-Komponenten, Systeme oder auch Anwendungen können und werden für die Erbringung unterschiedlicher Services benutzt. Kommunikation findet beispielsweise über dieselben Netzwerkkomponenten statt; Systeme werden für die Erbringung mehrerer Services genutzt was zu Redundanzen führt (Abbildung 19).

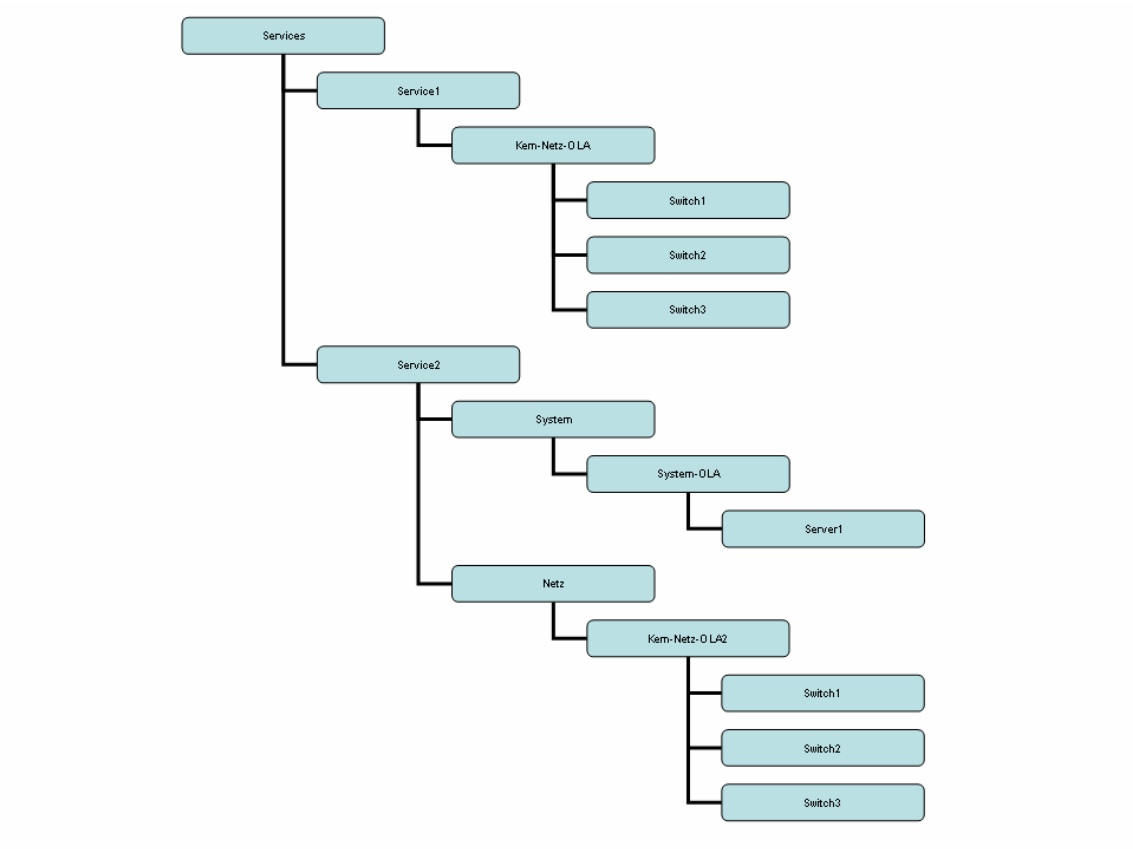

Abbildung 19: Service-Baum mit Redundanz

Um diese Redundanz zu vermeiden, sollen auch Services als Ressourcen aufgeführt werden können. Damit können bereits erfasste Services als Vorraussetzung für andere Services abgebildet werden.
Entscheidend ist dabei, dass bei der Modellierung in Service-NSM eine hinreichend genaue Abgrenzung der IT-Services gegeneinander, damit eine solche Vereinfachung überhaupt zum Tragen kommen kann.

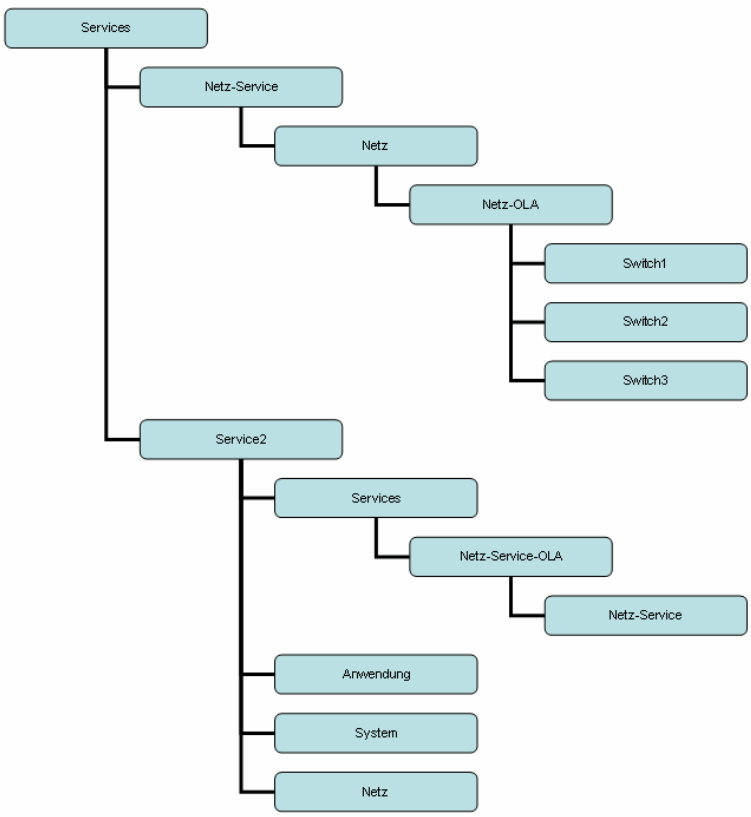

Abbildung 20: Servicebaum ohne Redundanz

4.3.2 Instanzen des Service-Objekts: Services, Ebenen, OLAs

Das hier vorgestellte Service-Objekt ist ein Container-Objekt und generisch in dem Sinne, dass es nicht nur als Wurzel-Objekt der Servicedarstellung dient, sondern auf mehreren Hierarchieebenen Anwendung findet.
Daher beschränkt das vorgeschlagene Datenmodell nicht die Modellierung eines Services derart, dass nur die hier beschriebene Struktur eines *Service Specification Sheets* möglich wäre; auch andere Strukturen ließen sich problemlos umsetzen. Es ist Aufgabe von Service-NSM, dem Benutzer bei der Erfassung der Daten im jeweiligen Kontext sinnvolle Optionen anzubieten.
Hier soll nur eine mögliche Umsetzung beschrieben werden, für die ein solches Datenmodell gut geeignet wäre.

4.3.3 Kopplung an Cable-NSM

Da die hier vorgeschlagene neue Funktionalität in der Plattform NSM umgesetzt werden soll, kann sie auch mit anderen Teilen von NSM kooperieren. Nutzbar für eine Service-Darstellung sind vor allem die im Komponenten-Baum von Cable-NSM enthaltenen Informationen.
Durch das Service-NSM soll ersichtlich sein, welche Komponenten zur Erbringung welcher IT-Services beitragen. Daher soll eine Verknüpfung zwischen Service-Objekt und Komponente stattfinden können.
Abbildung 21 zeigt eine Baumstruktur, bei der ein Ast einen Teil des Komponentenbaums wiedergibt, ein zweiter Ast einen ersten Schritt hin zu einem Servicebaum.

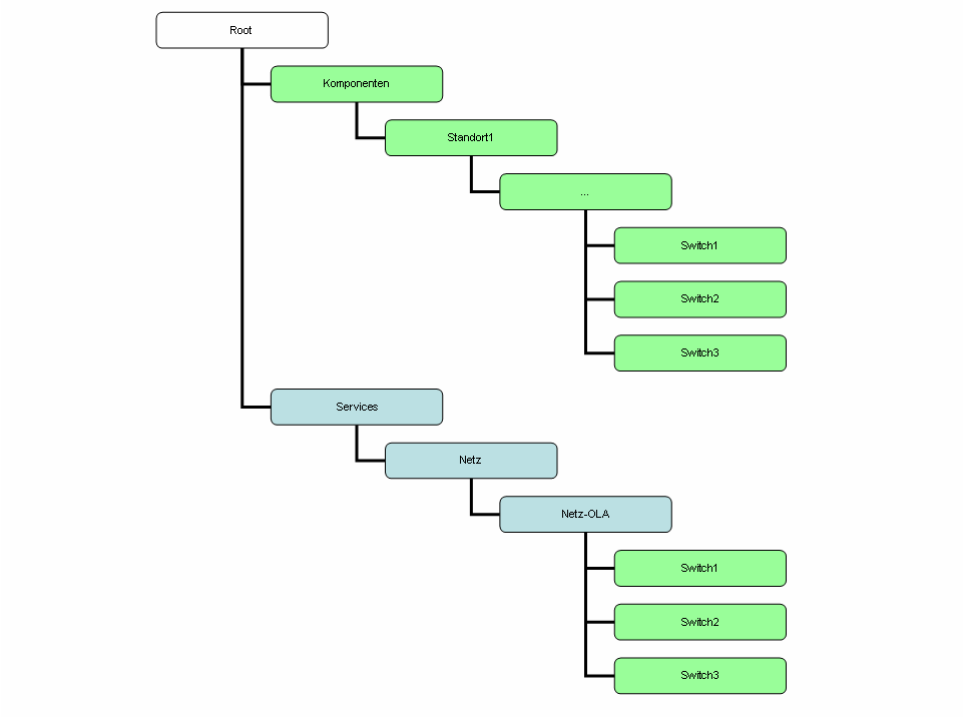

Abbildung 21: Komponenten- und Servicebaum

4.3.4 Operation Level Agreements und Underpinning Contracts im Service-Baum

Wichtiger Bestandteil eines *Service Specification Sheets* ist die Erfassung von Anforderungsparametern an die beschriebenen IT-Services. Diese werden in Vereinbarungen festgehalten; *Operation Level Agreements* (OLA) für solche innerhalb einer Organisation und *Underpinning Contracts* (UC) für solche mit externen Organisationen sind übliche Umsetzungen solcher Vereinbarungen; aus ihnen ergeben sich Anforderungen an die Infrastruktur, die diese erfüllen können muss.
Service-NSM kann OLAs und UCs als Verknüpfung zwischen IT-Service und dafür benötigter Infrastruktur nutzen. Diese werden zu eigenständigen Objekten im Service-Baum. Sie können Attribute erhalten, in denen beispielsweise Anforderungen an für die an sie gekoppelte Kind-Objekte (Komponenten, Services) erfasst werden können.

4.4 Umsetzung von Service-NSM

Um zu zeigen, dass eine Umsetzung dieser Vorschläge im Rahmen einer Management-Software wie NSM praktikabel und sinnvoll ist, wird hier ein Vorschlag vorgestellt, wie sich dieses verwirklichen ließe. Dabei wird die bestehende Struktur von NSM berücksichtigt; die Veränderungen sind umsetzbar, ohne existierende Datenbestände verändern zu müssen.

4.4.1 Einführung des Service-Begriffs in NSM

Das in Kapitel 4.1.4 eingeführte *Service Specification Sheet* wird als zusätzliche Baumstruktur analog der des Komponentenbaums von Cable-NSM eingeführt. Beginnend mit einem Wurzelknoten *Services* kann dann die oben dargestellte Service-Beschreibung direkt umgesetzt werden.
Zwei zusätzliche Anforderungen ergeben sich für das neue Service-Element im Servicebaum:

- Verknüpfungen mit Komponenten im Komponentenbaum sind über Assoziationen aus dem Servicebaum möglich
- Verknüpfungen von einem Service-Objekt zu einem anderen Service-Objekt im Servicebaum sind möglich

4.4.2 Modellierung des Servicebaums

Eine Umsetzung des Servicebaums in einem Service-NSM kann so aussehen, dass folgende neue Objekte und Assoziationen eingeführt werden:

- Service: Generisches Objekt, von dessen Typ die Knoten in Service-NSM instantiiert werden; solcherart sind Wurzelknoten, Service-Knoten (z.B. E-Mail, Kern-Netz usw.), Aufteilung in Ebenen (Services, Anwendungen, Systeme, Netz) und als Oberklasse für *Operation Level Agreements* und *Underpinning Contracts*
- ServiceTyp: Kategorie eines Service-Objekts (z.B. Service, Anwendung, System)
- ServiceAttribut: Beliebige zusätzliche Informationen über das Service-Objekt
- OperationsLevelAgreement: Modellierung einer Abmachung über Zusicherungen für Qualitätseigenschaften für weitere, damit verknüpfte Objekte innerhalb der gleichen Organisation
- UnderpinningContract: Modellierung einer Abmachung über Zusicherungen für Qualitätseigenschaften für externe Objekte

Diese lassen sich, wie in Abbildung 22 zu sehen, direkt auf bereits bestehende Datenstrukturen von Cable-NSM abbilden. Eine Veränderung des Datenbankschemas ist notwendig für die Umsetzung des Service-Objekts.

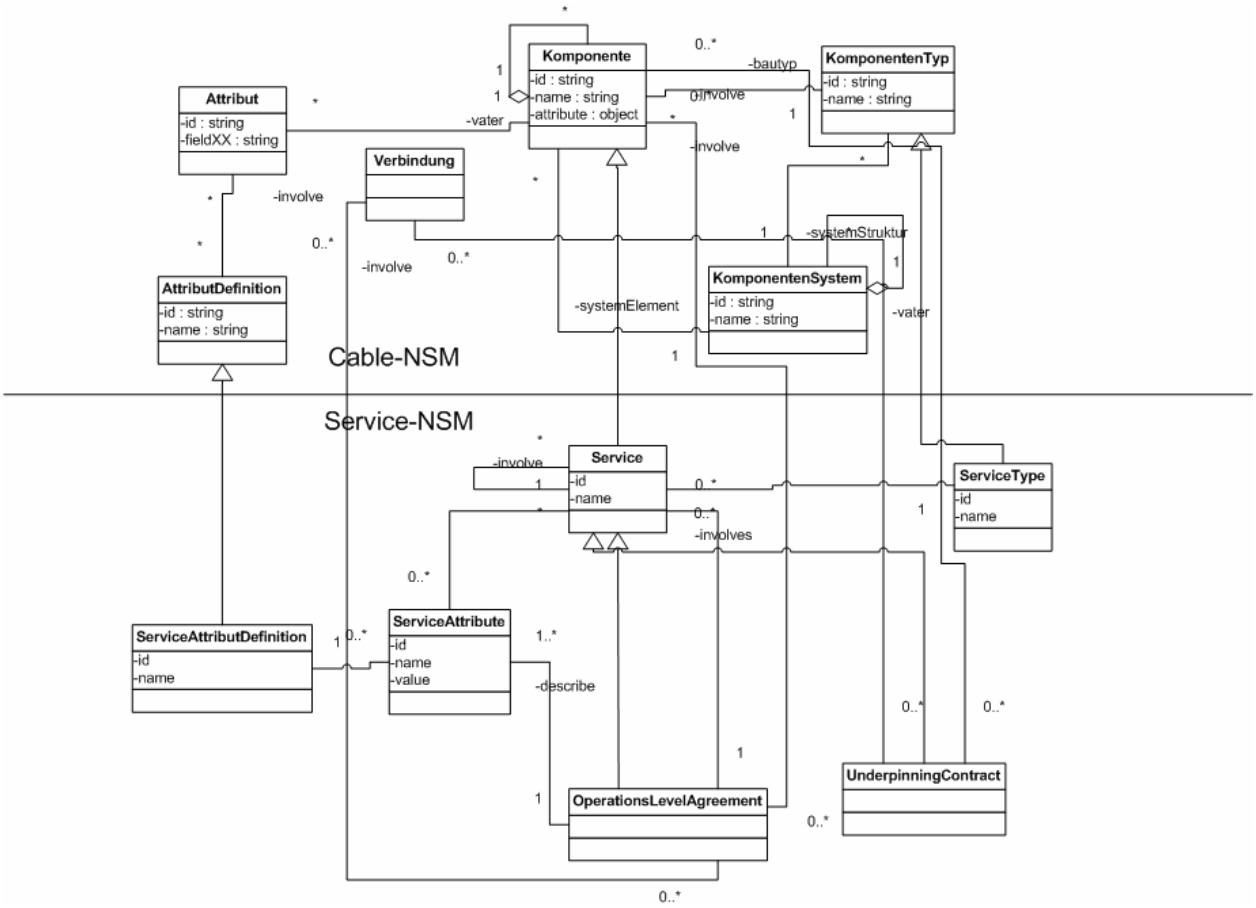

Abbildung 22: Klassendiagramm Service-NSM

4.4.3 Erweiterung des Datenbankschemas

Für die Abbildung der Assoziation zwischen Service-Objekten und davon Abhängigen Service-Objekten (1-zu-n-Beziehung) ist eine zusätzliche Tabelle notwendig (Tabelle 1). Hiermit lässt sich die Assoziation eines Service-Objekts zu weiteren Service-Objekten an beliebiger anderer Position im Servicebaum erfassen.

Tabelle 1: ServiceDependencies

Spaltenname	Typ	Null?
service_id	VARCHAR2(18)	Nein
involves_service_id	VARCHAR2(18)	Nein

4.4.4 Integration von Service-NSM in die Benutzeroberfläche von NSM

Die hier beschriebene Datenstruktur eines Servicebaums lässt sich ähnlich der des Komponentenbaums in einer Baumansicht unterbringen. Da sich die vorgeschlagene Datenstruktur eng an der vom Komponentenbaum anlehnt, können dieselben GUI-Elemente verwendet werden. Für den erfahrenen Benutzer ergibt sich so eine leichtere Einarbeitung, da er bereits mit der Arbeitsweise von Cable-NSM vertraut ist.

4.4.5 Umsetzung eines Servicebaums für den E-Mail-Dienst

Das Beispielszenario, anhand dessen Cable-NSM und Logic-NSM in Kapitel 3 vorgestellt werden, soll nun als Basis für einen exemplarischen Service-Baum dienen.
In Abbildung 23 ist eine exemplarische Umsetzung des E-Mail-Dienstes der ATIS als Servicebaum in Service-NSM zu sehen. Ausgehend von einem Wurzelknoten *Services* werden vier Services dargestellt: E-Mail, DNS, LDAP und Kern-Netz. DNS, LDAP und Kern-Netz sind die Basisdienste, die für die Erbringung des E-Mail Dienstes notwendig sind. Die für diese genutzte Hardware wurde in 1.3.2 vorgestellt.
Der Servicebaum als Umsetzung eines Service Specification Sheets kann die sich für diesen Dienst ergebenen Abhängigkeiten wiedergeben. Es wird erfasst, welche Teildienste für den E-Mail-Dienst notwendig sind, welche Komponenten der Infrastruktur beteiligt sind, und es werden beispielhaft einige OLAs eingebracht, in denen Anforderungen an beteiligte Teildienste und Komponenten erfasst werden können.

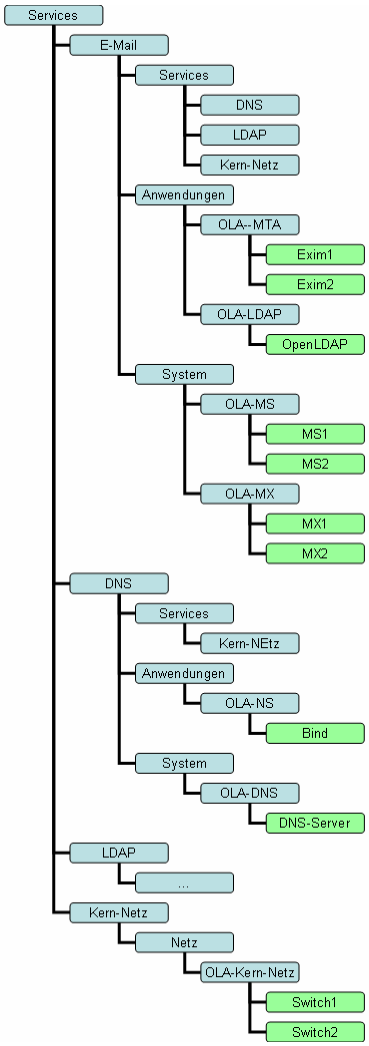

Abbildung 23: Servicebaum für den E-Mail-Dienst der ATIS

4.5 Demonstrator der Fa. Steinmayr

Im Rahmen der Arbeit an dieser Studienarbeit wurde von der Fa. Steinmayr ein Demonstrator implementiert, der hier ausgeführte Ideen aufgreift und umsetzt. Im Ergebnis gibt es zwischen beiden Ansätzen große Ähnlichkeiten, aber auch einige Unterschiede. Der hier vorgestellte Demonstrator ist eine Rückportierung von Funktionalität aus einer bis jetzt unveröffentlichten Version 7. Applikation und Datenmodell der hier verwendeten Version 6 wurden angepasst, um dieses zu ermöglichen.

Die Software selbst hat noch keinen Namen erhalten und ist nicht voll funktionsfähig. In diesem Kapitel wird sie deshalb Sensor-NSM genannt. Zum jetzigen Zeitpunkt ist sie aber hinreichend gut benutzbar, um die Umsetzung der hier gewünschten Funktionalität zu beschreiben.

Die Funktionalität, die der Demonstrator bereits zum jetzigen Zeitpunkt bietet, ähnelt der in Kapitel 4.4 beschriebenen. Schwerpunkte und Einsatzzweck unterscheiden sich allerdings. Die grundlegenden Unterschiede werden daher gesondert beschrieben.

4.5.1 Sensor-NSM

Kern der Umsetzung ist ein neuer Bestandteil von Logic-NSM (Abbildung 24). Ihm sind zwei neue Funktionen zugeordnet worden: Eine öffnet den Sensorbaum, die andere ein Fenster für die *Sensor-Historie*.
Der Sensorbaum (Abbildung 25) wird in einem Fenster dargestellt, das dem Komponentenbaum von Cable-NSM ähnelt. Es gibt einen Wurzelknoten (Root) und, davon absteigend, eine beliebig tief verzweigende Baumstruktur.
Wird ein neuer Knoten angelegt, erhält dieser einen beliebigen Namen; für den Typ des Knotens kann zwischen zwei Modi gewählt werden, indem das Feld Zusammenfassung aktiviert bleibt oder abgewählt wird. Ist *Zusammenfassung* aktiviert, bleibt der Knoten im Sensorbaum eigenständig, d.h. er ist unabhängig von den Objekten, die in Cable-NSM erfasst sind. Er stellt ein generisches Objekt im Sensorbaum dar.
Ist *Zusammenfassung* deaktiviert, kann in den Feldern *Klasse* und *Objekt* jeweils eine Auswahl getroffen werden. Das Feld *Klasse* bietet die Auswahlmöglichkeiten *Komponente*, *Verbindung* und *Nutzung*. Je nach Auswahl in diesem Feld kann im Feld *Objekt* ein Objekt aus Cable-NSM des jeweiligen Typs ausgewählt werden. Damit wird dieser Knoten direkt mit Cable-NSM verknüpft.

4.5.2 Erfassung eines Services im Sensorbaum

Knoten, bei denen *Zusammenfassung* deaktiviert ist, können damit für die Umsetzung eines *Service Specification Sheets* wie in 4.4.2 beschrieben genutzt werden. Wie in Abbildung 25 zu sehen ist, kann der Servicebaum vollständig abgebildet werden. Für die geforderte Verknüpfung zu Komponenten aus Cable-NSM wird *Zusammenfassung* deaktiviert und ein an das jeweilige Objekt gebundener Knoten erstellt.
Bis jetzt ist es nicht möglich, einen Knoten anzulegen, der eine Verknüpfung der Klasse *Sensor* o.ä. besitzt. Dieser würde erst die gewünschte Abhängigkeit eines Service von einem anderen ermöglichen.

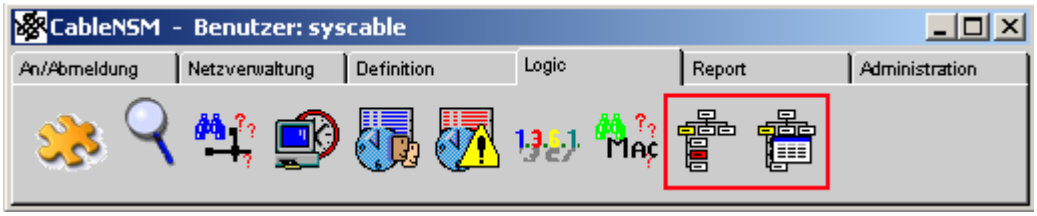

Abbildung 24: Neue Optionen im Hauptmenü für Sensor-NSM

4.5.3 Datenmodell des Sensorbaums

Für den Sensorbaum im Demonstrator ist das Datenmodell von NSM erweitert worden. Eine neue Tabelle in der Datenbank enthält die relevanten Daten und Verknüpfungen nach Cable-NSM. Das Datenbankschema der Tabelle *Sensortreeobject* ist in Tabelle 2 zu sehen.
Jeder Knoten bildet eine Zeile in der Tabelle. *Zusammenfassung* ist im Feld ISSUMMARY abgelegt. Wird eine Verknüpfung zu einem Objekt aus Cable-NSM angelegt, wird deren ID in OBJECTID gespeichert.

Tabelle 2: Schema der Tabelle "SENSORTREEOBJECT"

Spaltenname	Datentyp	Null?
ID	VARCHAR2(18)	Nein
OBJECTID	VARCHAR2(18)	Ja
ICONYELLOW	VARCHAR2(18)	Ja
ICONRED	VARCHAR2(18)	Ja
ISSUMMARY	VARCHAR2(1)	Ja

SOURCE	VARCHAR2(2000)	Ja
OBJECTTYPE	VARCHAR2(30)	Ja
ICONGREEN	VARCHAR2(18)	Ja
NAME	VARCHAR2(100)	Nein
STATUS	NUMBER	Ja
VATER	VARCHAR2(18)	Ja
WIZZARDARRAY	VARCHAR2(2000)	Ja

4.5.4 Zusätzliche Funktionalität

Wie in den Abbildungen und im Datenbankschema zu sehen, hat jedes Objekt des Sensorbaums eine Eigenschaft *Status*. Der Oberfläche nach zu urteilen gibt es die Möglichkeiten, entsprechend den Ampelfarben, *grün*, *gelb* und *rot*.
Im Fenster „Sensor-Historie“ lässt sich in einer Liste von Statusänderungen von Knoten im Sensorbaum verfolgen. Jede Änderung trägt einen Zeitstempel.
Für die Ermittlung des Wertes dieses Statusfeldes ist bereits ein (geplanter) Mechanismus erkennbar. Auf der rechten Seite des Dialogs werden *Regeln* angezeigt, die sie zu- und wegschalten lassen. Diese unterscheiden sich abhängig davon, ob *Zusammenfassung* an- oder abgeschaltet ist.
Für Knoten des Typs *Zusammenfassung* soll sich der aktuelle Statuswert aus den Statuswerten der Kind-Knoten ableiten; hierfür gibt es Bedingungen wie *höchster Statuswert* oder anteilig überwiegende Statuswerte. Für Knoten, die mit Objekten aus Cable-NSM verknüpft sind, kann der Statuswert aus Werten aus Attributfeldern der assoziierten Objekte abgeleitet werden. Für numerische Attributwerte können hierfür Vergleichsoperatoren wie ‚>’, ‚=’ usw. und konstante Faktoren genutzt werden.

4.5.5 Differenzen in der Ausrichtung der Ansätze

Mit dieser Status-Information zielt Sensor-NSM eindeutig auf Funktionalität, die eine Erweiterung von NSM in Richtung Monitoring der im Sensorbaum erfassten Services gehen soll.
Diese Stoßrichtung unterscheidet sich stark von der, die in Kapitel 4.4 vorgestellt wird. Der hier beschriebene Vorschlag dient ausschließlich der Erfassung und Dokumentation eines Service auf Basis einer Interpretation von *Service Specification Sheets*. Der Monitoring-Aspekt wird darin nicht aufgegriffen.

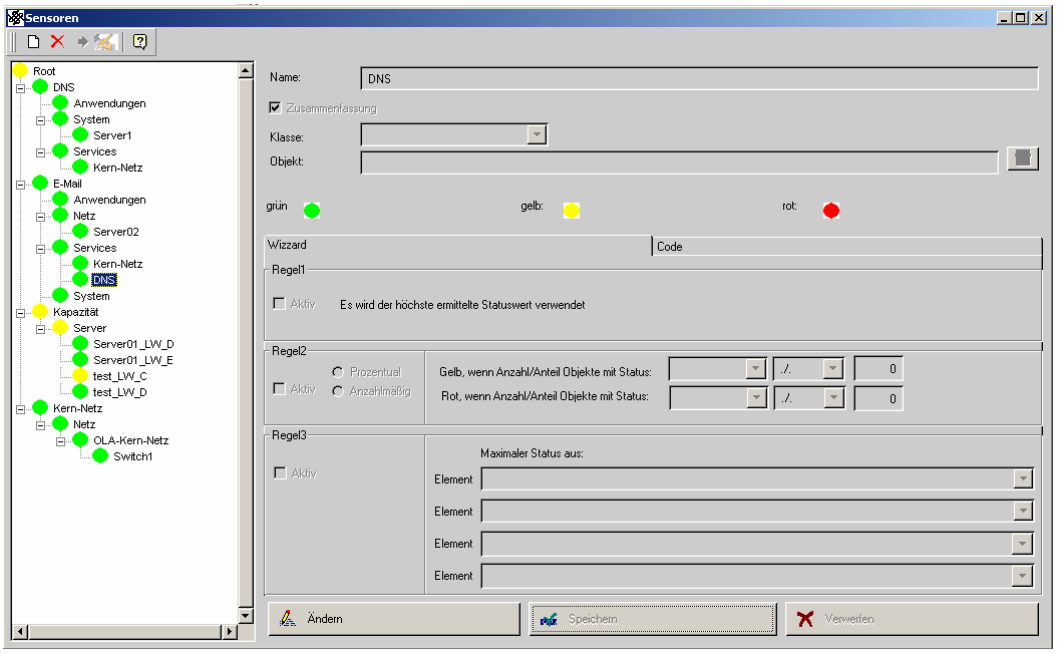

Abbildung 25: Sensorbaum

4.5.6 Alternative eines unabhängigen Service-NSM und Sensor-NSM

Durch die Verknüpfung von Monitoring und Servicebaum wird versucht, einen direkten Zusammenhang zwischen Service-Objekten und Daten aus dem Monitoring herzustellen.
Dieser Ansatz ist problematisch, da

- an Services automatisch die Erwartungshaltung geknüpft wird, diese seien überwachbar
- ein direkter Zusammenhang hergestellt wird zwischen der Funktion von Komponenten und der Bereitschaft, einen Dienst zum gegenwärtigen Zeitpunkt erbringen zu können
- die Komplexität, eingebracht durch OLAs und UCs übersprungen wird, in dem diese nicht vorgesehen sind bzw. deren Eigenschaften nicht berücksichtigt werden können; diese sind aber notwendig für eine Umsetzung von *Service Specification Sheets*

Es ist so keine hinreichende Abgrenzung möglich, welche Teile von NSM für welche Managementdomäne konzipiert sind. Gerade dies wäre aber notwendig, um NSM im Kontext von serviceorientiertem Management zu betrachten.
Aus diesen Gründen wird vorgeschlagen, Sensor-NSM separat von Service-NSM zu betrachten und zu implementieren. Eine separate Betrachtung würde Service-Spezifikationen wir den Servicebaum nicht mit domänenfremden Aspekten des Monitoring verbinden und ihnen diese damit aufzwingen.

5 ANALYSE VON NAGIOS UND IT-INFRASTRUKTURMONITORING

Die ATIS setzt die unter der *GNU General Public License* stehende Monitoringsoftware *Nagios* ein. Sie wurde im Jahr 1999 von dem US-Amerikaner Ethan Galstad zunächst unter dem Namen *Netsaint* veröffentlicht und wird seitdem unter seiner Führung weiterentwickelt. Inzwischen (April 2008) liegt die stabile Version 3.0.1 vor. Nagios unterstützt die System- und Netzwerkadministratoren, indem es die Überwachung von Infrastrukturkomponenten wie Netzwerkgeräten und Serverrechnern, der darauf laufenden Systemprozesse und Netzdienste (SMTP, POP3, HTTP etc.) und von weiteren Systemressourcen und -attributen wie beispielsweise CPU-Auslastung oder Länge einer Druckerwarteschlange ermöglicht. Darüber hinaus können mittels spezieller netzwerkfähiger Sensoren weitere technisch messbare Parameter wie Umgebungstemperatur oder Stromversorgung überprüft werden. Die ermittelten Statusinformationen werden auf einem übersichtlichen Webinterface dargestellt, bleiben aber auch zusätzlich in Logfiles oder der angeschlossenen Datenbank gespeichert, um weitergehende Auswertungen wie die Erstellung von Status-Historien und Verfügbarkeitsberichten über vergangene Zeiträume zu ermöglichen. Besondere Stärken besitzt Nagios im Bereich der, im Falle von Fehlern oder Ausfällen relevanten, Benachrichtigungsfunktionen, die vielseitig und tiefgehend konfigurierbar sind. Des Weiteren lassen sich so genannte *Event Handler* definieren, die beim Auftreten bestimmter Ereignisse automatisch in den Betrieb eingreifen: Zum Beispiel kann ein korrumpierter Serversystemprozess, der keine sinnvollen Daten mehr zurückliefert, auf diese Weise ohne menschliche Interaktion abgebrochen und neu gestartet werden.
In diesem Kapitel wird zunächst die grundlegende Struktur von Nagios beschrieben und dann detaillierter auf einige im weiteren Verlauf dieser Arbeit interessante Komponenten und deren Funktionsweise eingegangen. Insbesondere das Informationsmodell wird genauer untersucht, da es später wieder aufgegriffen und in verschiedenen Ansätzen erweitert wird. Es wird versucht, die inhaltlichen Überschneidungen zu den Erläuterungen in [Pan07] gering zu halten, manchmal sind sie jedoch zugunsten eines besseren Verständnisses der Zusammenhänge von Vorteil.

5.1 Struktur und Funktionsweise von Nagios

Bei der Software Nagios handelt es sich ein modular aufgebautes *Monitoring-Framework*. Zur Erfassung von Statusdaten folgt es einem Manager-Agent-Prinzip: Die Kernanwendung hat hauptsächlich organisatorische und koordinierende Aufgaben. Der eigentliche Funktionsumfang wird erst durch eine große Anzahl von verfügbaren Erweiterungen erreicht. Mit deren Hilfe können viele unterschiedliche Typen von Statusdaten ermittelt werden, andere rüsten spezielle Funktionen zur Datenauswertung, -darstellung und Benachrichtigung nach.
Die modulare Struktur bietet weitere Vorteile. So ermöglicht sie es, Nagios mit geringem Aufwand mit anderen spezialisierten Managementwerkzeugen (z. B. MRTG, RRDtool) zu verknüpfen.
Detailliertere Informationen über die Architektur von Nagios sind unter anderem in [Gal07a] und [Pan07] zu finden. Eher praktisch orientierte Ausführungen zur Installation und Konfiguration von Nagios enthält [Rie06].

5.1.1 Zentrale Komponenten

Auf einem Monitoring-Host mit Unix-Betriebssystem läuft der zentrale Nagios-Daemon. Dieser enthält eine so genannte *Check Logik*, die für den Aufruf der Plugins, die dann die eigentlichen Statusprüfungen durchführen, zuständig ist. Weiterhin kümmert sich der Daemon um das Einlesen und Prüfen der Konfiguration, die Auswertung und Sicherung der Statusdaten, deren Aufbereitung für das Webinterface und das Verschicken von Benachrichtigungen, welches wiederum über Plugins realisiert ist. [Rie06]
Alternativ zu einer lokalen Speicherung von Statusdaten in Logfiles kann eine Datenbankanbindung über das so genannte *NDOMOD-Eventbrokermodul* realisiert werden. Damit ist Nagios in der Lage, neben den Monitoringdaten auch die vollständige Konfiguration

in einer SQL-Datenbank abzulegen. Dies birgt Vorteile in Bezug auf die Möglichkeiten zur Verarbeitung und Wiederauffindbarkeit der gespeicherten Daten und dient außerdem als Grundlage für die geplante Entwicklung eines neuen PHP-basierten Webinterface. [Gal07b]
Die erwähnten Komponenten und deren funktionale Beziehungen zueinander sind noch einmal in Abbildung 26 dargestellt.

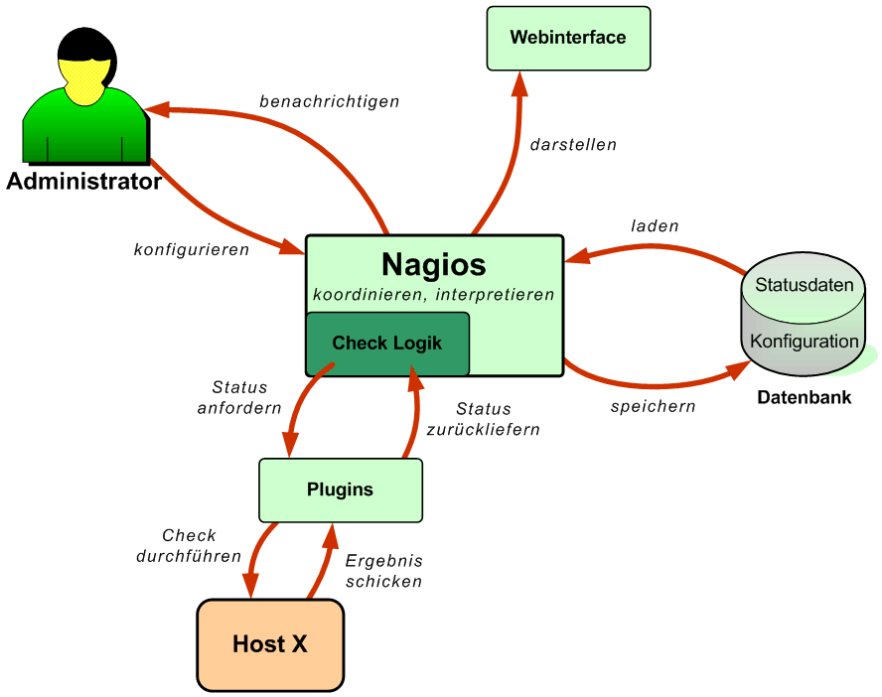

Abbildung 26: Komponenten und deren Beziehungen

5.1.2 Nagios-Informationsmodell

Aus den Dokumentationen zu Nagios und dem NDOUtils Datenbankmodell (siehe [Gal07a] und [Gal07c]) lassen sich Informationen zu den in Nagios verwendeten Objekttypen und deren Relationen, also ein Nagios-Informationsmodell ableiten. Es existiert eine relativ große Anzahl unterschiedlicher Objekttypen. Neben den zu überwachenden Entitäten gibt es Typen zur Definition von Zeitintervallen, Fehlerfällen und Kontakten für die Benachrichtigung.

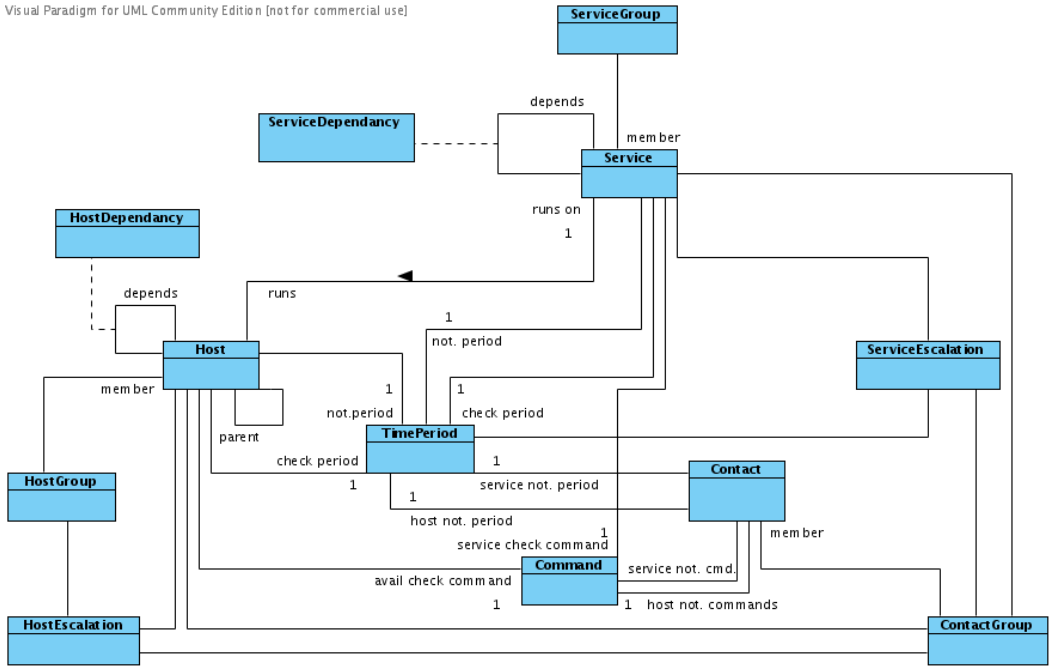

Abbildung 27: Klassendiagramm der Objekttypen in Nagios [Pan07]

Da in dieser Arbeit die interne Steuerung des Monitorings, die Interpretation der Statusdaten und die Abläufe zur Benachrichtigung nicht weiter interessieren, wird das bereits in [Pan07] erstellte Klassendiagramm aus Abbildung 27 um die nicht relevanten Klassen reduziert. Konkret fallen die Typen *TimePeriod*, *Command*, *Contact*, *ContactGroup*, *HostEscalation* und *ServiceEscalation* aus der Betrachtung heraus. Übrig bleiben die Klassen *Host*, *HostGroup*, *HostDependency*, *Service*, *ServiceGroup* und *ServiceDependency*. Das resultierende vereinfachte Diagramm ist in Abbildung 28 zu sehen und wird im Folgenden detailliert erklärt.

Hosts & Services

Alle zu überwachenden Objekte werden in Nagios entweder als *Host* oder als *Service* klassifiziert.

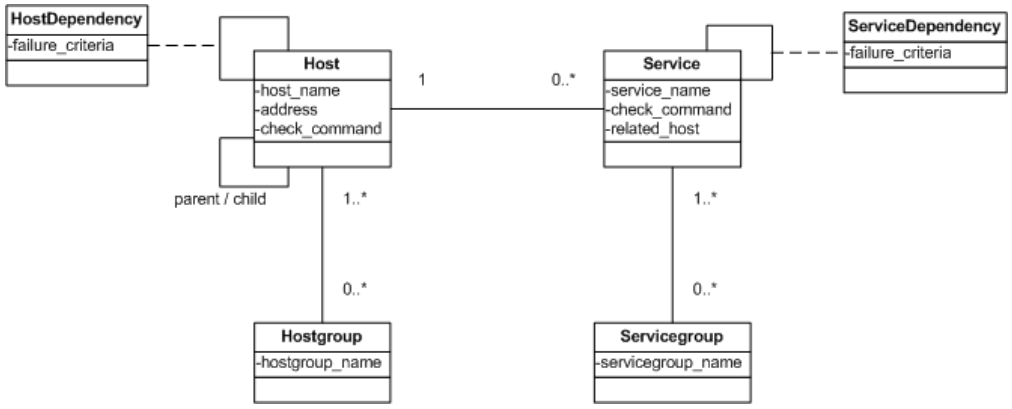

Abbildung 28: Vereinfachtes Klassendiagramm der Objekttypen

Ein *Host* ist ein (zumeist physikalisch vorhandenes) Objekt, das über eine Adresse, üblicherweise eine IP-Adresse, identifiziert wird. Es handelt sich also um die Repräsentation von Servern, Arbeitsplatzrechnern und virtuellen Maschinen aber auch von Netzinfrastrukturkomponenten wie Routern und Switchen. Jeder Host besitzt eine Prüfroutine (*host_check*), mir dem der allgemeine Verfügbarkeitsstatus ermittelt werden kann. Dies

geschieht meistens über einen simplen ICMP-Echo-Request (ping). Der Objekttyp Host besitzt zudem noch eine reflexive parent/child-Relation, mit deren Hilfe eine Baumstruktur über alle Geräte mit dem Nagios-Host als Wurzel aufgebaut werden kann. Anders gesagt entsteht ein Erreichbarkeitsgraph, durch den unnötige Benachrichtigungen unterdrückt werden können. Ist ein Host A also nicht erreichbar, weil auf dem Pfad dort hin bereits ein fehlerhaft arbeitendes Netzwerkgerät (Host B) von Nagios erkannt wurde, so wird für Host A der Status UNREACHABLE gesetzt und fortan keine Benachrichtigungen verschickt.
In die Klasse *Service* fallen alle übrigen Objekte, die von Nagios überwacht werden können, beispielsweise Netzdienste, Systemprozesse oder -ressourcen eines Hosts. Sie sind im Gegensatz zu den Hosts durchweg nicht physikalisch fassbar. Für jeden Service muss genau ein bereits existierendes Hostobjekt benannt sein, auf dem er sich befindet; ohne Host kann es in Nagios also keine Services geben. Umgekehrt kann ein Host jedoch beliebig viele Services beherbergen. Analog zum *host_check* ist auch für jeden Service ein Prüfbefehl (*service_check*) definiert, mit dem der aktuelle Status in Erfahrung gebracht werden kann. Hier kann der Administrator aus dem vollen Arsenal der *Nagios-Checks* (siehe nachfolgender Abschnitt 5.1.4) schöpfen.
Hosts können in so genannten *Hostgroups*, Services in *Servicegroups* zusammengefasst werden. Dies dient dem Vereinfachen der Konfiguration von ähnlichen Objekten und bietet eine einfache aber auch eingeschränkte Möglichkeit zur Strukturierung von Objekten, die logisch zusammengehören. Bei der Konfiguration von Gruppen ist zu beachten, dass jeder Hostgroup / Servicegroup je mindestens ein Host / Service zugewiesen sein muss, damit sie existieren kann. Jeder Host kann in beliebig vielen Hostgroups / jeder Service in beliebig vielen Servicegroups vertreten sein. Hosts können jedoch nicht Mitglied von Servicegroups werden und umgekehrt.
Die Assoziationsklassen *HostDependency* und *ServiceDependency* ermöglichen die Definition von Abhängigkeiten zwischen Hosts bzw. Services. Dieser strukturierende Mechanismus dient der Abbildung von Beziehungen zwischen Objekten der Infrastruktur und der Unterdrückung von unnötigen Checks und Benachrichtigungen.
Es ist offensichtlich, dass der Service in Nagios nichts mit dem in Kapitel 2.1 definierten Begriff des IT-Service zu tun hat. Um Verwechselungen vorzubeugen, wird deshalb bis zum Ende dieses Kapitels ‚Service' nur für den Nagios-Service und ‚IT-Service' oder ‚Dienst' für den von einem IT-Infrastrukturbetreiber erbrachten Service benutzt.

5.1.3 Zustände

Aus der Sicht von Nagios besitzt jedes überwachte Objekt zu jedem Zeitpunkt genau einen der folgenden für sie gültigen Zustände (Status):

Tabelle 3: Zustände (Status) in Nagios

Host	**UP**		**DOWN**	**UNREACHABLE**	**PENDING**
Service	**OK**	**WARNING**	**CRITICAL**	**UNKNOWN**	**PENDING**

Es fällt auf, dass es für Hosts keinen „mittleren" Status gibt, vergleichbar mit WARNING bei den Services. Dies liegt darin begründet, dass ein *host_check* immer entweder funktioniert (UP), fehlschlägt (DOWN) oder, mangels funktionierenden Pfads zu dem Host, gar nicht durchgeführt werden kann (UNREACHABLE). Ein teilweise erfolgreich durchgeführter *host_check* und ein damit erforderlicher Zwischenzustand existieren also nicht.
service_checks liefern hingegen meistens einen gemessenen numerischen Wert zurück, der nach den vorher vom Administrator festgelegten Schwellwerten von der Check Logik interpretiert wird. Theoretisch könnten somit sehr viele Zustände zwischen OK und CRITICAL definiert werden; Nagios kennt – nicht zuletzt um die Komplexität gering zu halten – aber nur die Zwischenstufe WARNING.
Der Status PENDING wird in dem Fall verwendet, dass ein Check bisher noch keine Rückmeldung geliefert hat, aber ein Timeout noch nicht überschritten wurde.

5.1.4 Checks

Die Funktion zum Prüfen eines Statusdatums wird bei Nagios allgemein als *Check* bezeichnet. Es stehen insgesamt drei Möglichkeiten zur Verfügung, Checks durchzuführen (siehe Abbildung 29):

(1) Der Nagios-Prozess kann einen Check direkt ausführen. Dies funktioniert nur, falls sich der angesprochene Host in einem Netzwerksegment befindet, den Nagios direkt erreichen kann. Außerdem können auf diese Weise nur die generelle Verfügbarkeit eines Geräts (*host_check*, 1a) und von ‚außen' erreichbare Netzdienste geprüft werden (1b). Beispielsweise kann der Füllstand einer Festplatte auf diese Weise nicht ermittelt werden.

(2) Um an Statusdaten zu kommen, die ein Host nicht über das die Netzwerkschnittstelle nach außen sichtbar macht, müssen auf diesen Geräten so genannte Agenten eingesetzt werden. Sie enthalten, ähnlich wie der Nagios-Hauptprozess, eine Check Logik, über die lokal installierte Check Plugins aufgerufen werden können. Nagios bedient sich eines speziellen Agenten-Checks, der dem entfernten Agenten Prüfaufträge zukommen lässt und erhält über den gleichen Kanal die gewünschten Werte zurück.

(3) Die dritte Möglichkeit besteht darin, auf die unaufgeforderte Übermittlung von Statusinformationen zu warten. Auf den entfernten Hosts müssen dazu selbständige Programme oder Skripte eingerichtet sein, die im Falle einer Statusänderung unaufgefordert aktiv werden und eine Nachricht an den Nagios-Host verschicken. Nagios enthält zu diesem Zweck eine Schnittstelle, die auf solche Anforderungen von außen reagiert (*check acceptor*).

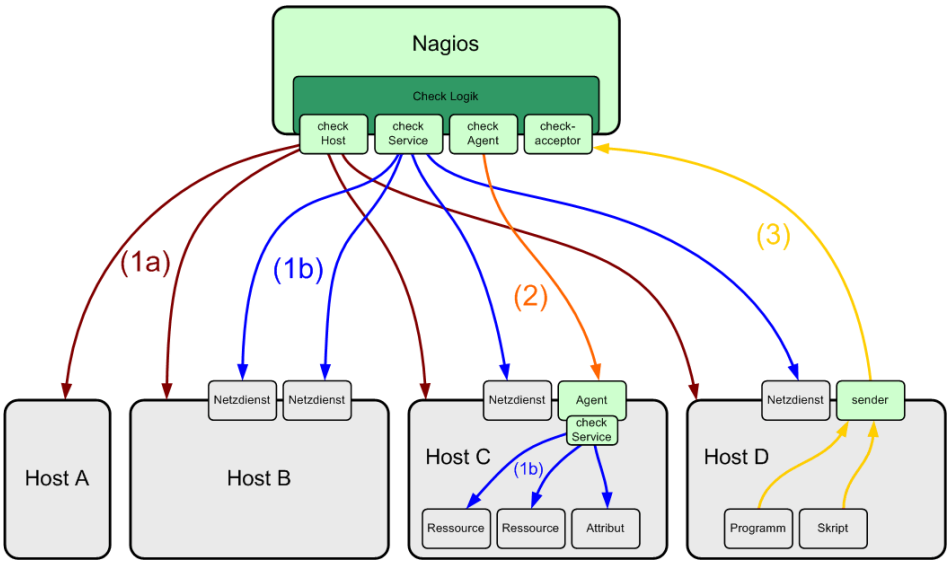

Abbildung 29: Verschiedene Typen von Checks

Die ersten beiden Varianten gehören zur Klasse der so genannten *Active Checks*, also üblicherweise regelmäßigen und von Nagios aktiv ausgeführten Prüfungen. Variante drei beschreibt im Gegensatz dazu einen *Passive Check*: Nagios stößt hierbei die Checks nicht direkt an, sondern wartet darauf, dass sich die überwachten Hosts von selbst melden.

5.1.5 Plugins

Plugins erweitern die Fähigkeiten von Nagios, indem sie neue Typen von Checks nachrüsten. Eine Vielzahl von Plugins ist kostenlos erhältlich, darüber hinaus können sie auf recht einfache Weise für eigene Anwendungszwecke selbst programmiert werden.

Die Check Logik des Nagios-Prozesses ruft, wie in Abbildung 30 dargestellt, das für einen Check erforderliche Plugin auf, das dann den aktuellen Status des Host oder Service ermittelt und zurückliefert.
Eine weit verbreitete Plugin-Sammlung sind die so genannten *Nagios Plugins*. Sie enthalten eine große Anzahl von häufig eingesetzten Checks, beispielsweise *check_http* zum Überwachen eines Webservers oder *check_disk*, das Informationen über den verbleibenden Speicherplatz liefert.

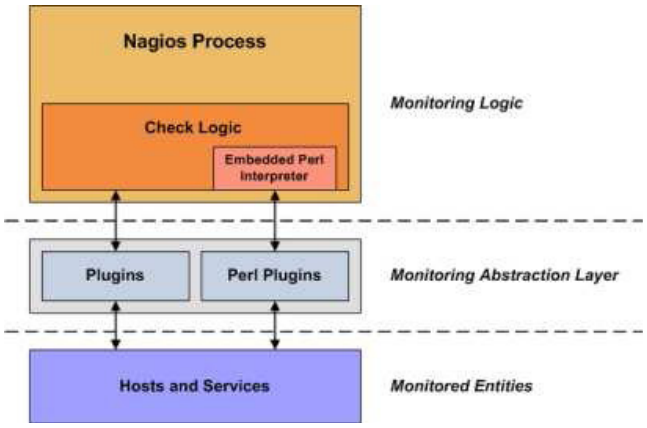

Abbildung 30: Funktionsweise von Nagios-Plugins [Gal07a]

5.1.6 Agenten (NRPE, NSCA)

Agenten stellen die verlängerten Arme des Nagios-Host dar, da dieser selbst nicht alle Checks direkt ausgeführt kann. Der NRPE (Nagios Remote Plugin Executor) und der NSCA (Nagios Service Check Acceptor) sind zwei der am häufigsten eingesetzten Agenten für Nagios.

NRPE

Das *Nagios Remote Plugin Executor* (NRPE) Addon ermöglicht *Active Checks* vom Typ (2). Es besteht aus zwei Teilen: Das *check_nrpe* Plugin befindet sich auf dem Nagios-Host, der *NRPE-Daemon* wird direkt auf dem entfernten Linux/Unix-System installiert, das überwacht werden soll. Hier lauscht der Daemon dann auf Anfragen vom Nagios-Hauptprozess.
Zur Ermittlung von Statusdaten ruft Nagios zunächst lokal das *check_nrpe* auf und teilt ihm mit, welcher Service-Check auf welchem Host durchgeführt werden soll. Dieses baut dann eine optional SSL-gesicherte Verbindung zum TCP-Port 5666 des Remote Host auf und übergibt dem entfernten NRPE-Daemon die Anfrage (siehe Abbildung 31). Dieser führt nun seinerseits ein auf dem überwachten Host lokal installiertes Plugin aus, das die gewünschten Statusdaten ermittelt und sie an den Daemon zurückliefert. Sie fließen dann weiter zurück über die TCP-Verbindung an das *check_nrpe* bzw. den Nagios-Prozess. [Gal07d]

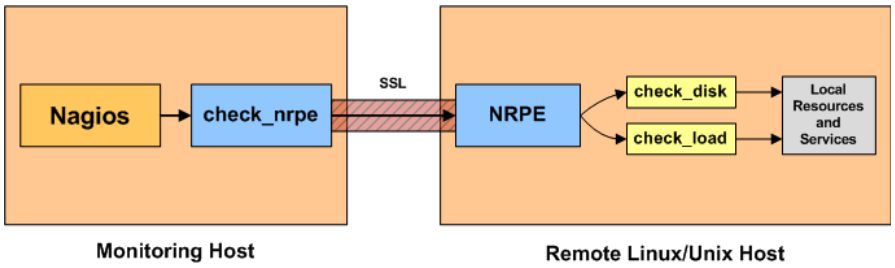

Abbildung 31: Funktionsweise des NRPE [Gal07d]

NSCA

Der *Nagios Service Check Acceptor* (NSCA) basiert dagegen auf dem Prinzip des Passive Check (Typ 3). Dazu wartet er auf von den überwachten Hosts selbständig – also ohne

Anforderung vom Nagios-Prozess – versandte Statusdaten. Der NSCA wird auf dem Nagios-Host installiert und lauscht nach erfolgter Konfiguration auf dem TCP-Port 5667. (siehe Abbildung 32)

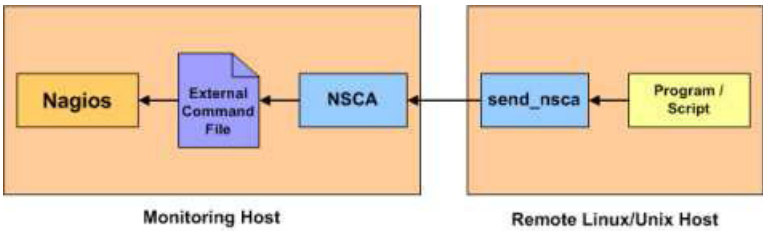

Abbildung 32: Funktionsweise des NSCA [Gal07a]

5.2 Infrastrukturmonitoring mit Nagios

Die aktuelle Version 3.0 von Nagios wird seit kurzer Zeit in der ATIS eingesetzt. Die aktiven Netzinfrastrukturkomponenten der ATIS wurden vollständig als Hosts eingepflegt, ebenso werden die meisten Serverrechner auf Verfügbarkeit geprüft. Auch viele wichtige Services werden auf diesen Geräten von Nagios überwacht.

5.2.1 NagVis-Karten

Um die Übersichtlichkeit der Darstellung von Statusinformationen über das Webinterface zu erhöhen, wurden mit dem Nagios-Addon *NagVis* strukturierte Karten eingepflegt. Sie lehnen sich an die physikalische Dokumentation des LINK-Netzes (siehe http://www.atis.uka.de/964.php) an und stellen überwiegend die Geräte der Netzinfrastruktur, also Switches, Router und Firewalls, in Standort-, Gebäude- bzw. Etagenansichten dar (siehe Abbildung 33).

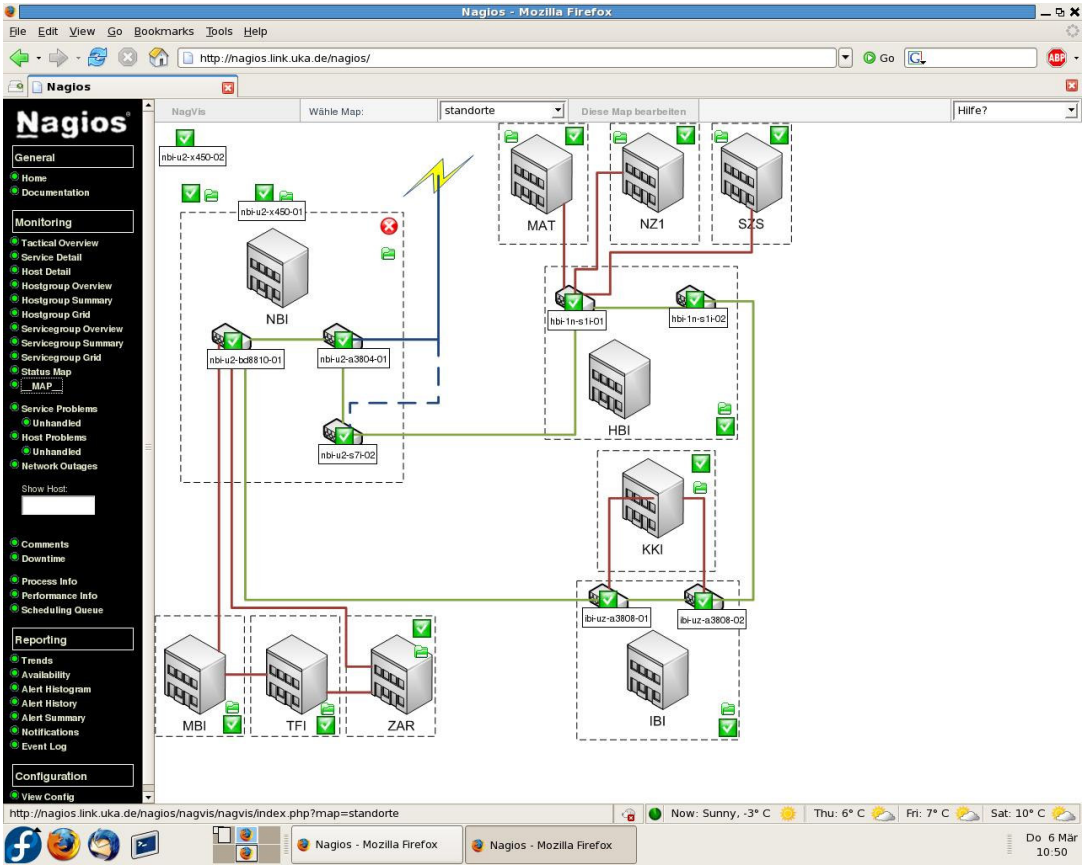

Abbildung 33: NagVis-Karte „Standorte“

Die Konfigurationsoberfläche von NagVis integriert sich nahtlos in das Webinterface von Nagios, ist also bequem per Browser erreichbar. Sie erlaubt das Anlegen, Modifizieren und Löschen von Karten auf denen bereits definierte Objekte beliebig positioniert werden können. Dazu zählen neben den Hosts, Hostgroups, Services und Servicegroups auch andere, schon vorhandene NagVis-Karten. Alle Objekte werden durch kleine Piktogramme repräsentiert; sie sind standardmäßig bei Hosts und Hostgroups quadratisch, bei Services und Servicegroups rund und NagVis-Karten werden durch ein kleines Ordnersymbol repräsentiert. Der Status der Objekte wird durch die Farbe der Symbole kenntlich gemacht:

- Grüner Haken (UP / OK): Der Status des Objekts ist in Ordnung. Alle Checks konnten erfolgreich durchgeführt werden / die Statuswerte sind im normalen Bereich.
- Gelber Blitz (WARNING): Das Objekt befindet sich in einem kritischen Zustand. Es ist zwar noch erreichbar und arbeitet ordnungsgemäß, ist jedoch über eine vorher definierte Schwelle ausgelastet.
- Rotes Kreuz (DOWN / CRITICAL): Der Host ist nicht erreichbar / der Service ist überlastet.
- Graues Fragezeichen (UNREACHABLE / UNKNOWN): Der Status des Objekts kann nicht ermittelt werden bzw. ist nicht bekannt.
- Grün-rotes Baustellensymbol (ACKNOWLEDGE): Der Status des Objekts wurde von einem Administrator wahrgenommen und quittiert. Es werden derzeit keine Checks durchgeführt.

Mit NagVis erstellte Karten werden in den folgenden Betrachtungen zur Serviceorientierung von Monitoringsystemen nochmals aufgegriffen.

6 SERVICE SPECIFICATION SHEET FÜR EIN MONITORINGSYSTEM

Die in Monitoringsystemen überwachten Objekte liegen dort häufig zunächst unstrukturiert vor. Zum Teil existieren einfache Methoden zum Zusammenfassen von technisch ähnlichen oder anderweitig zusammengehörigen Elementen, die Darstellung von komplexen IT-Services ist jedoch oft, so auch bei der Software Nagios, nicht vorgesehen.
Um ein Monitoringsystem in den Kontext von serviceorientiertem IT-Management zu stellen ist es daher erforderlich, den IT-Service innerhalb der Domäne des Monitorings greifbar zu machen. Als Ausgangspunkt dazu dient das *Service Specification Sheet* eines IT-Service. Anhand der damit repräsentierten statischen Service-Struktur sollen die für das Monitoring wichtigen, zu überwachenden Objekte identifiziert werden, welche für die Erfassung von dynamischen Aspekten des Service relevant sind.
In diesem Kapitel wird zu diesem Zweck das bereits in Kapitel 4 ausgearbeitete Konzept des Servicebaums aufgegriffen. Die daraus entwickelte Struktur wird dann auf das Beispielszenario E-Mail-Dienst angewendet, um den Ansatz zu veranschaulichen.

6.1 Servicebaum für das Monitoring

In Kapitel 4.3 wurde für die Repräsentation der statischen Aspekte des *Service Specification Sheet* eines IT-Service in einem *Asset-/Configuration*-Managementsystem der Servicebaum entworfen. Dass sich dieser jedoch auch als Grundlage für die Ableitung der für ein Monitoringsystem relevanten Elemente eignet, soll im Folgenden begründet werden.

6.1.1 Struktur des Servicebaums

Die Darstellung des Servicebaums ist in Abbildung 34 zu sehen. Er orientiert sich an der Struktur aus Abbildung 23, jedoch sind hier aus Gründen der Übersichtlichkeit OLAs und UCs nicht aufgeführt.
Er besitzt am oberen Ende zunächst einen Wurzelknoten namens ‚Services', an dem alle weiteren Objekte verzweigen. Auf der zweiten Ebene befinden sich alle bisher modellierten IT-Service-Objekte, inklusive des beispielhaften *Services X*, der hier genauer betrachtet wird. Der interessante Teil des Servicebaums befindet sich auf Ebene drei. Hier findet eine Unterteilung der an der Bereitstellung und Erbringung eines IT-Service beteiligten Komponenten in verschiedene Kategorien statt:

- Unter ‚Services' können vollständige, bereits modellierte IT-Dienste der zweiten Ebene referenziert werden. Dies trägt der Tatsache Rechnung, dass für den Betrieb von Service X das Funktionieren von anderen Services vorausgesetzt werden muss. Man spart sich mit dieser Kategorie das wiederholte, also redundante Definieren von immer wieder benötigten Basisdiensten (wie z. B. DNS-Dienst) bis hinunter auf die einzelnen Komponenten.
- In die Kategorie ‚Software' fallen alle Arten von Anwendungen, Systemprozessen, Netzdiensten, Treibern und sonstigen Programmen.
- Bei ‚Systeme' werden alle Hosts, Server, Virtual Machines einsortiert. Auch deren Bestandteile (CPU, Festplatte, etc.) und Eigenschaften gehören dazu.
- Unter ‚Netzwerk' fallen alle Objekte, die zusammen das Datennetz darstellen. Dazu gehören sowohl die aktiven, als auch die passiven Komponenten, aus denen sich das Datennetz zusammensetzt. Also neben den Routern, Switchen und Firewalls auch alle verwendeten Kabel, Patchpanels, Kupplungen etc.
- Im Bereich ‚Andere Komponenten / Non-IT-Services' werden sonstige Objekte erfasst, die zur Erbringung des Service beitragen. Als Beispiel seien hier Stromversorgung/ USV, Klimatisierung und erforderliches Wartungspersonal genannt.

Auf Ebene vier sind schließlich die konkreten Objekte dieser Kategorien zu erkennen.

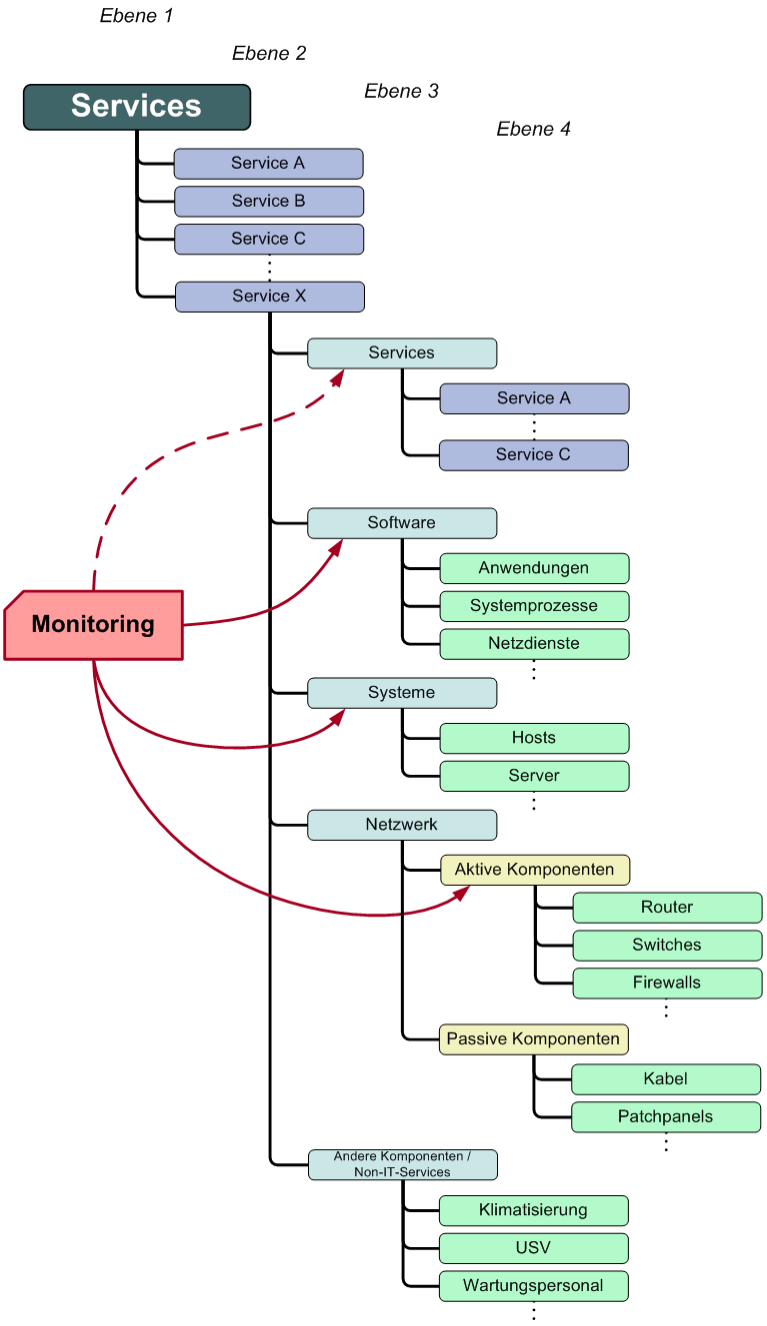

Abbildung 34: Servicebaum für ein Monitoring-System

6.1.2 Eignung für das Monitoring

Zunächst stellt sich die Frage, ob der Servicebaum überhaupt für die Benutzung in einem Monitoringsystem taugt. Dazu muss er folgende Anforderung erfüllen: Alle Objekte, die von einem Monitoring überwacht werden können, müssen auch in dem Baum vertreten sein.
Dass dies der Fall ist, kann leicht damit erklärt werden, dass sich der Servicebaum in der gleichen Struktur bereits für die Verwendung in einem Asset-Management-System qualifiziert hat, somit die an einem Service beteiligten Objekte und damit auch die für das Monitoring relevanten Objekte enthalten sind.

6.1.3 Reduzierter Servicebaum für das Monitoring

Das Konzept des Servicebaums wird nun herangezogen, um davon abzuleiten, welche Komponenten für einen Service X im Rahmen eines serviceorientierten Managements überwacht werden müssen.
Ein pragmatischer Ansatz wäre es, einfach alle Objekte des Service X im Servicebaum zu monitoren. Hier stößt man jedoch schnell auf die Einschränkung, dass sich gar nicht alle Objekte monitoren lassen. Beispielsweise können für passive Netzwerkkomponenten wie Kabel und Patchpanels keine Statusinformationen (im Sinne eines technischen Monitorings) ermittelt werden.
Es muss also eine Auswahl getroffen werden, was überhaupt in die Überwachung aufgenommen werden kann. Die roten Pfeile, die in Abbildung 34 von dem Kasten „Monitoring“ wegführen, zeigen die von einem klassischen Monitoringsystem überwachbaren Objekte. Dazu zählen die aktiven (adressierbaren) Netzwerkkomponenten, die Systeme und die Software. Die Services gehören auch dazu, jedoch nur mittelbar, da sie sich wiederum rekursiv den drei anderen Bereichen zusammensetzen. Sie gelten nur dann als überwachbar, falls sie bereits als Dienste angelegt wurden.

6.2 Beispielszenario E-Mail-Dienst

An dieser Stelle wird das Beispielszenario E-Mail-Dienst aufgegriffen. Es wird zunächst zusammengefasst, welche Objekte darin enthalten sind und diese anschließend kategorisiert. Einen guten Überblick gibt die in Abbildung 35 noch einmal dargestellte Infrastruktur des Kern-Maildienstes.

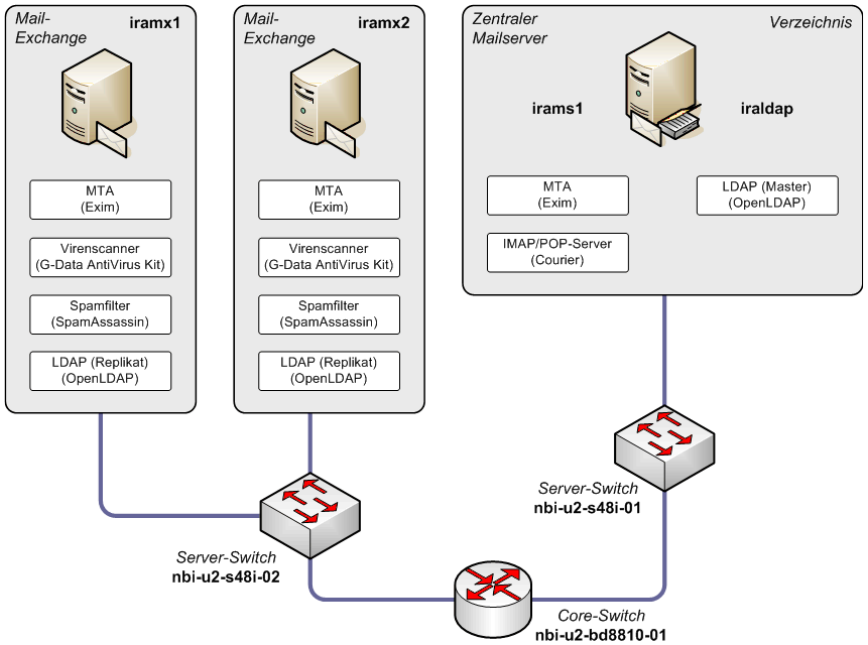

Abbildung 35: Infrastruktur des Kern-Maildienstes

6.2.1 Beteiligte Komponenten

An der Erbringung des Kern-E-Mail-Dienstes sich folglich die in Tabelle 4 aufgelisteten Komponenten beteiligt:

Tabelle 4: Komponenten des Maildienstes

Server	zugehörige Software / Systemprozesse
iramx1	OpenLDAP, Exim, G-Data AntiVirus, SpamAssassin
iramx2	OpenLDAP, Exim, G-Data AntiVirus, SpamAssassin
irams1 / iraldap	OpenLDAP, Exim, Courier

Switches
nbi-u2-bd8810-01
nbi-u2-s48i-01
nbi-u2-s48i-02

Hinzu kommt noch der DNS-Dienst, der, wie in Abschnitt 1.3.2 erläutert, ebenfalls zum Core-Mailservice gehört.

6.2.2 Kategorisierung der Komponenten

Entsprechend den Kategorien des Servicebaums aus Abschnitt 6.1.1 werden nun die Komponenten sortiert:

Tabelle 5: Kategorien der Komponenten

Services
DNS-Dienst

Software
OpenLDAP (3x)
Exim (3x)
G-Data AntiVirus (2x)
SpamAssassin (2x)
Courier (1x)

Systeme
Server iramx1
Server iramx2
Server irams1 / iraldap

Netzwerk (aktive Komponenten)
Core-Switch nbi-u2-bd8810-01
Server-Switch nbi-u2-s48i-01
Server-Switch nbi-u2-s48i-02

6.2.3 Monitoring-Servicebaum für das Beispielszenario

Die vorsortierten Komponenten werden im nächsten Schritt in den für ein Monitoringsystem angepassten, also reduzierten Servicebaum eingefügt. Es ergibt sich die folgende Struktur:

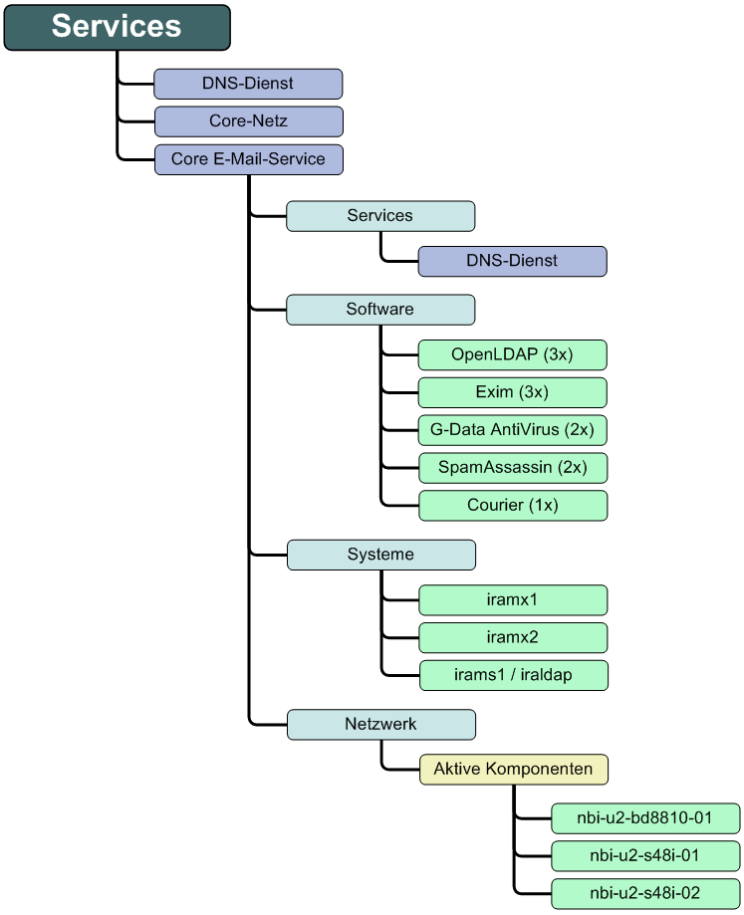

Abbildung 36: Reduzierter Monitoring-Servicebaum für den E-Mail-Service

Es wurde gezeigt, dass das Konzept des Servicebaums für die Ermittlung der zu überwachenden Objekte einer IT-Infrastruktur (Messpunkte), in Bezug auf einen bestimmten IT-Service, geeignet ist. Das Ergebnis ist der daraus entwickelte, für das Monitoring reduzierte Servicebaum, der eine Strukturierung der als relevant identifizierten Objekte darstellt.

7 NAGIOS ALS TEIL EINES SERVICEORIENTIERTEN IT-MANAGEMENTS

Nach der Ableitung der für die Erfassung der dynamischen Aspekte eines IT-Service relevanten Monitoring-Informationen von IT-Ressourcen mit Hilfe des Konzepts des Servicebaums, sollen nun in einem Best-Practise-Verfahren Möglichkeiten untersucht werden, wie mit der Software Nagios diese Daten sinnvoll erfasst und, auf den IT-Dienst bezogen, strukturiert werden können. Sie kann damit Teil einer Architektur für serviceorientiertes IT-Management werden (Abbildung 37).

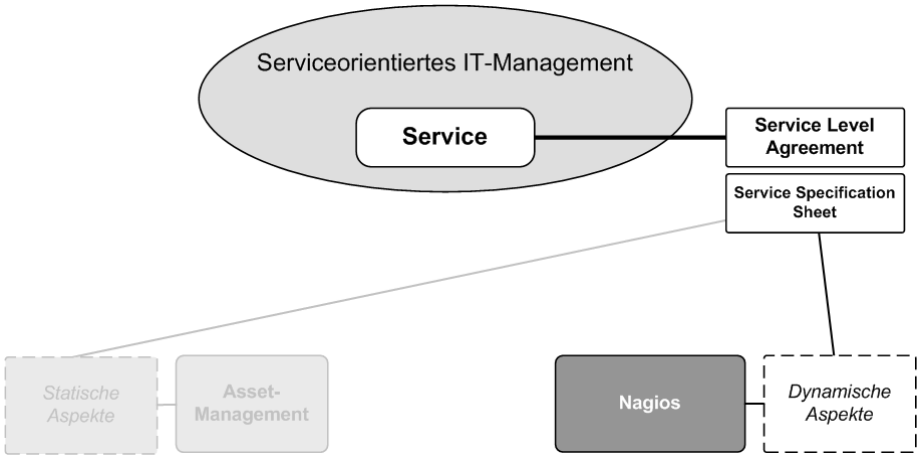

Abbildung 37: Nagios als Teil eines serviceorientierten Managements

In diesem Kapitel werden vier verschiedene Ansätze vorgestellt, wie Nagios in eine solche Architektur integriert werden kann: Zuerst wird mit den bordeigenen Mitteln von Nagios und dem Addon NagVis die Darstellung des Servicebaums realisiert (7.1). Der zweite Ansatz macht von einer Modifikation des Nagios-Informationsmodells Gebrauch (7.2). Den Service direkt in Nagios zu implementieren, wird ab hier nicht weiter verfolgt.
Stattdessen wird eine Integration der Monitoringsoftware in ein leistungsfähigeres und standardisiertes Managementsystem vorgeschlagen (7.3). Der folgende Ansatz 3 beschäftigt sich daher mit der Frage, wie die Informationen aus der Nagios-Datenbank für ein *Web-Based Enterprise Management* (WBEM)-System genutzt werden können (7.4). Ansatz 4 schließlich stellt die Rolle des Nagios-Hauptprozess im Rahmen einer serviceorientierten Managementarchitektur infrage und untersucht die Möglichkeit, direkt bei den Nagios-Agenten auf den Managed Hosts die Statusinformationen für ein WBEM-System abzugreifen (7.5).
Die Vor- und Nachteile eines jeden Ansatzes werden erörtert. Teilweise wird das Beispielszenarios E-Mail-Dienst herangezogen, um die Zusammenhänge zu veranschaulichen.

7.1 Ansatz 1 – Native Abbildung des E-Mail-Dienstes in Nagios

Der E-Mail-Dienst wird ausschließlich mit Hilfe der vorhandenen Möglichkeiten von Nagios modelliert. Zur Visualisierung kommt zusätzlich das Addon NagVis zum Einsatz. Das Ergebnis soll grafisch ansprechend und übersichtlich dargestellt werden.
Auf einer Service-Übersichtskarte zeigt ein Symbol jeweils den Gesamtstatus eines einzelnen Dienstes an. Falls weitere Informationen über die mit einem Dienst verbundenen Komponenten und deren Struktur gewünscht sind, führt ein Klick auf das Symbol zu einer detaillierten Servicekarte. Diese orientiert sich in ihrer Strukturierung und Darstellung an dem für ein Monitoringsystem reduzierten Servicebaum aus dem vorigen Kapitel: Der E-Mail-Dienst als NagVis-Karte ist in Abbildung 38 zu sehen.

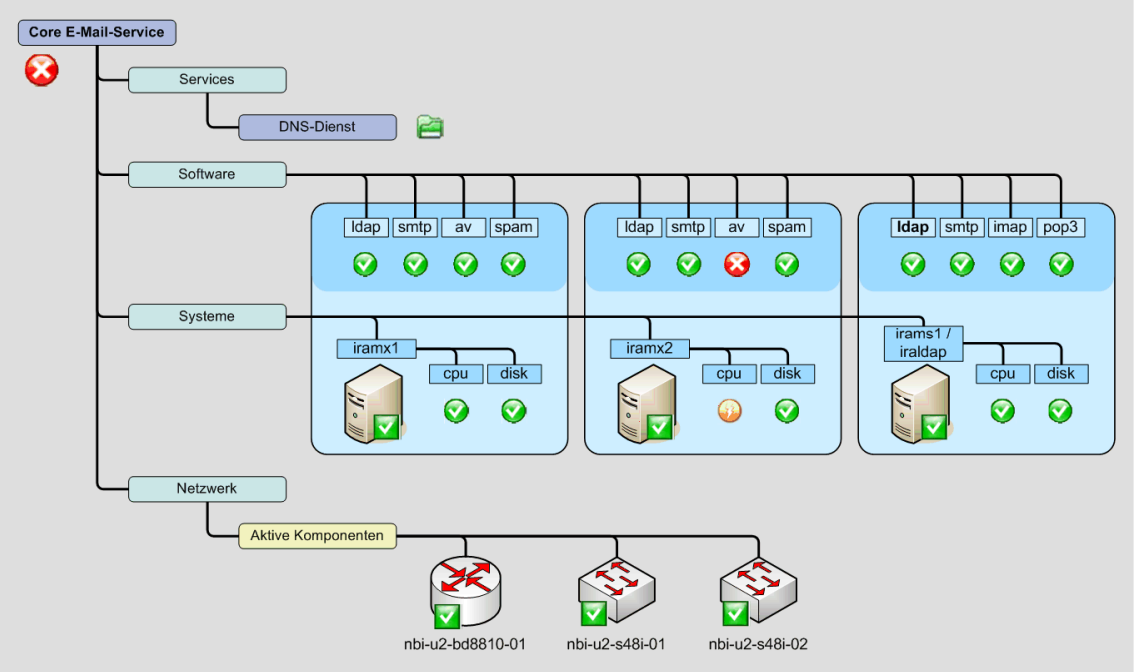

Abbildung 38: Der E-Mail-Dienst dargestellt in einer NagVis-Karte

Es ist möglich, die Erweiterung der Funktionalität durch das Addon NagVis mit Hilfe eines modifizierten Informationsmodells darzustellen. Die NagVis-Karten können als ein neuartiger Objekttyp aufgefasst werden, der mit keinem der bisher vorhandenen Nagios-Objekttypen deckungsgleich ist oder von einem solchen vererbt werden kann. Die Erweiterung des Informationsmodells ist in Abbildung 39 dargestellt. Das ursprüngliche Klassenmodell ist in schwarz gehalten, zur besseren Unterscheidung wurde dessen Klassennamen ein ‚Nagios_' vorangestellt. Änderungen und neu hinzugefügte Teile des Diagramms sind in roter Farbe eingezeichnet.

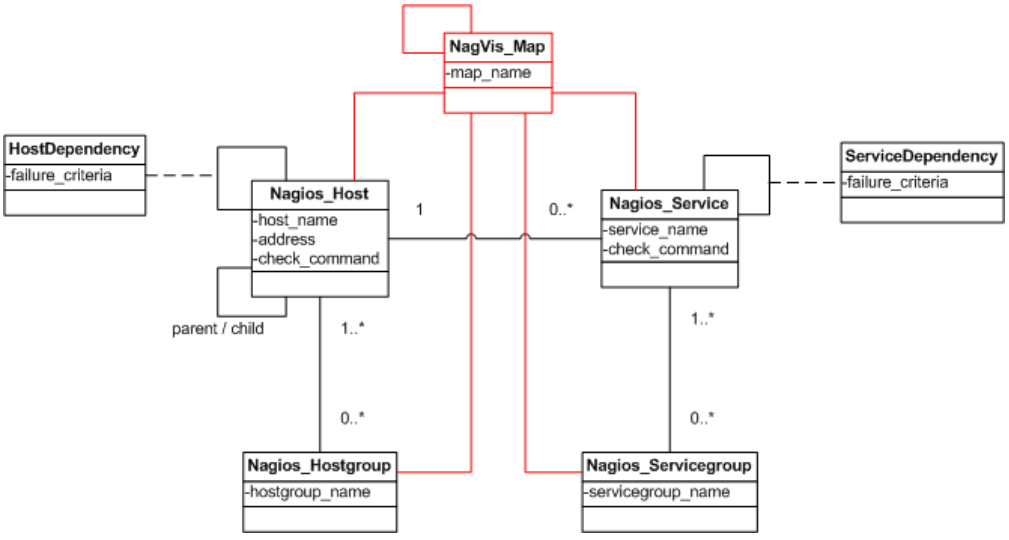

Abbildung 39: Modifikation des Informationsmodells durch NagVis

Die neue Klasse *NagVis_Map* hat Assoziationen zu den bisherigen Klassen *Host*, *Service*, *Hostgroup*, *Servicegroup*, da auf einer NagVis-Karte beliebige Kombinationen von diesen Objekten angelegt werden können. Außerdem besitzt sie eine reflexive Assoziation, da eine Karte auch andere Karten als Objekte enthalten kann.

7.1.1 Bewertung von Ansatz 1

Vorteile

Die Erstellung einer NagVis-Karte ist mit geringem Arbeits- und Zeitaufwand möglich. Auch Veränderungen am Setup können über das komfortable Konfigurationsinterface problemlos eingepflegt werden.
Die Karten bieten dem Administrator eine relativ simple, übersichtliche Perspektive über die Zugehörigkeit von Komponenten zu einem Service und deren Status. Durch die freie Wahl des Bildhintergrundes kann eine nahezu beliebige Strukturierung, im Beispiel eine an den reduzierten Servicebaum angelehnte, grafisch realisiert werden.
Die implizite Erweiterung des Informationsmodells durch NagVis bringt (eingeschränkte) neue Möglichkeiten mit sich, wie Objekte zueinander in Relation gesetzt werden können. Beispielsweise kann ein Host zusammen mit einem Service in einer Gruppe aufgenommen werden. Dies ist mit den Methoden von Nagios ohne das NagVis-Addon nicht möglich.

Nachteile

Die Strukturierung, die durch NagVis-Karten vorgenommen wird, beschränkt sich auf die Zusammenfassung der hinzugefügten Objekte in einer neuen Gruppe. Ein weiteres In-Beziehung-Setzen ist nicht möglich bzw. beschränkt sich auf die Benutzung einer speziellen Hintergrundgrafik und ist daher rein optischer Natur. Es können daher keine Aussagen darüber getroffen werden, auf welche Weise sich das Fehlverhalten einer Komponente auf die Verfügbarkeit und Qualität eines Dienstes auswirkt.
Der simulierte überlastungsbedingte Ausfall der AntiVirus-Komponente auf dem Server iramx2 (siehe Abbildung 38) verdeutlicht diesen Zusammenhang: Da sich somit (mindestens) ein Objekt der NagVis-Karte im Zustand DOWN / CRITICAL befindet, wechselt auch der Gesamtstatus des E-Mail-Service auf CRITICAL (Abbildung 38 links oben). Dass in Wirklichkeit durch die Redundanz der Server iramx1 und iramx2 der Dienst weiterhin reibungslos funktioniert, kann bei der Modellierung mit NagVis nicht abgebildet und daher auch nicht dargestellt werden.
Eine Zusammenführung dieses Ansatzes mit anderen Management-Tools mit dem Ziel, die Serviceorientierung eine integrierten Managements zu unterstützen, ist nicht oder nur mit erhöhtem Aufwand möglich.

7.2 Ansatz 2 – Der IT-Service im Nagios-Informationsmodell

In diesem Ansatz wird untersucht, ob sich mittels einer Erweiterung des Nagios-Informationsmodells ein IT-Service als Objekttyp neu einführen lässt.

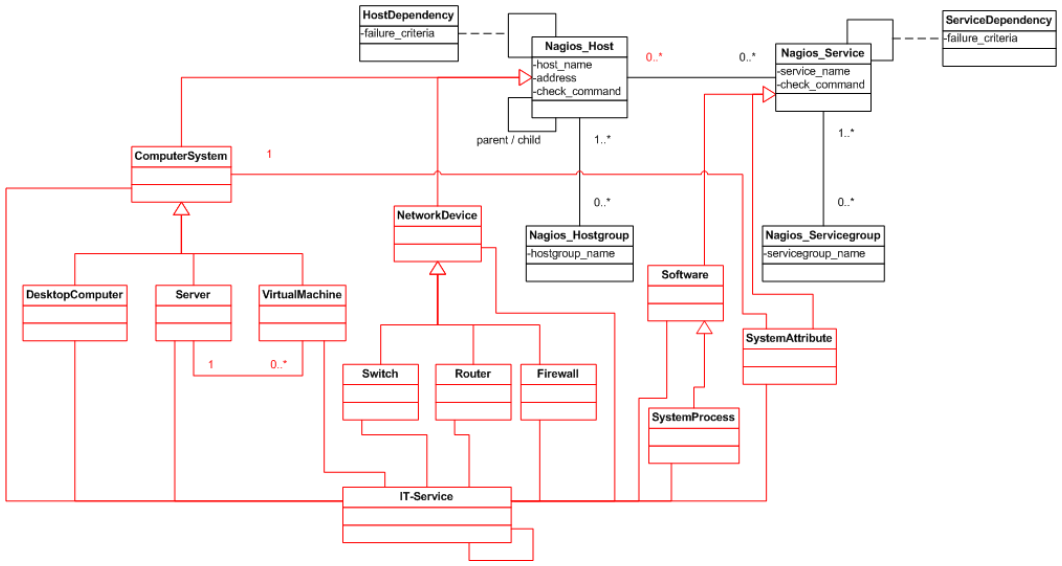

Abbildung 40: Erweitertes Nagios-Informationsmodell

Um zur Abbildung eines IT-Service zu gelangen, wie er durch die Struktur des reduzierten Servicebaums vorgeschlagen wurde, müssen dessen Kategorien Services, Software, Systeme und Netzwerk vorhanden sein. Sie sind im Nagios-Informationsmodell bisher nicht enthalten und werden daher angelegt. Die Klasse *ComputerSystem* wird von dem Nagios-Objekttyp Host vererbt, ebenso wie die Klasse *NetworkDevice*. Hier ist jedoch eine Besonderheit zu beachten: In der Kategorie Systeme des Servicebaums sind neben den Rechnern nämlich auch noch deren Attribute, wie beispielsweise CPU-Auslastung, untergebracht. Letztere vererben sich in Gestalt der Klasse *SystemAttribute* vom Nagios-Objekttyp Service, ebenso wie die Klasse *Software*.
Die Klasse *IT-Service* ist der Kern der Modellerweiterung. Sie komponiert sich aus den anderen vier neuen Klassen und außerdem über eine reflexive Assoziation aus sich selbst.

7.2.1 Bewertung von Ansatz 2

Vorteile

Die Modellierung des IT-Service im Nagios-Informationsmodell bietet eine hohe Flexibilität, da im Prinzip die komplette Struktur des Servicebaums neu hinzugefügt werden muss. Strukturen und Abhängigkeiten können deutlich präziser erfasst werden als mit Ansatz 1. Damit lassen sich, mittels der vorliegenden Statusdaten, gegebenenfalls auch qualitative Aussagen zu den Eigenschaften eines Dienstes machen.

Nachteile

Es handelt sich bei der Erweiterung des Nagios-Informationsmodells um eine proprietäre Entwicklung. Bei veränderten Anforderungen in der Zukunft muss die Modellierung immer wieder selbst angepasst werden. Änderungen am Nagios-Datenmodell in neueren Versionen von Nagios können Probleme/ Nacharbeit verursachen. Unter Umständen ergeben sich Inkompatibilitäten im Zusammenspiel von neuen Releases mit dem modifizierten Informationsmodell.
Durch den proprietäre Basis dieses Ansatzes ändert sich im Vergleich zu Ansatz 1 nichts an der schlechten Integrierbarkeit in weiter gefasste serviceorientierte Managementumgebungen.

7.3 Nagios als Zulieferer eines WBEM-Systems

Wie die Bewertungen der Ansätze 1 und 2 gezeigt haben, sind die Möglichkeiten der Modellierung eines abstrakten IT-Service innerhalb von Nagios entweder sehr begrenzt oder sehr aufwändig zu realisieren und zu warten.

Es ergibt sich daher eine Motivation, die Modellierung des Service in ein dafür besser geeignetes, flexibleres und leistungsfähigeres Managementsystem zu verlegen. Dies birgt das Potential, ausgehend von den ermittelten Statusdaten, auch qualifizierte Aussagen über die Verfügbarkeit und die Qualität eines IT-Dienstes machen zu können.
Die Auswahl einer Standard-Architektur bietet außerdem die Möglichkeit, andere verfügbare Managementtools mit entsprechenden Schnittstellen in ein serviceorientiertes Gesamtsystem zu integrieren. Als standardisierte Architektur bietet sich WBEM an.
Nagios soll im Weiteren auf seine Funktionen als spezialisiertes Monitoring-Tool für IT-Komponenten beschränkt bleiben. Im Rahmen eines serviceorientierten Managements kann es als Zulieferer von aktuellen Monitoringdaten an ein WBEM-System dienen (siehe Abbildung 41).

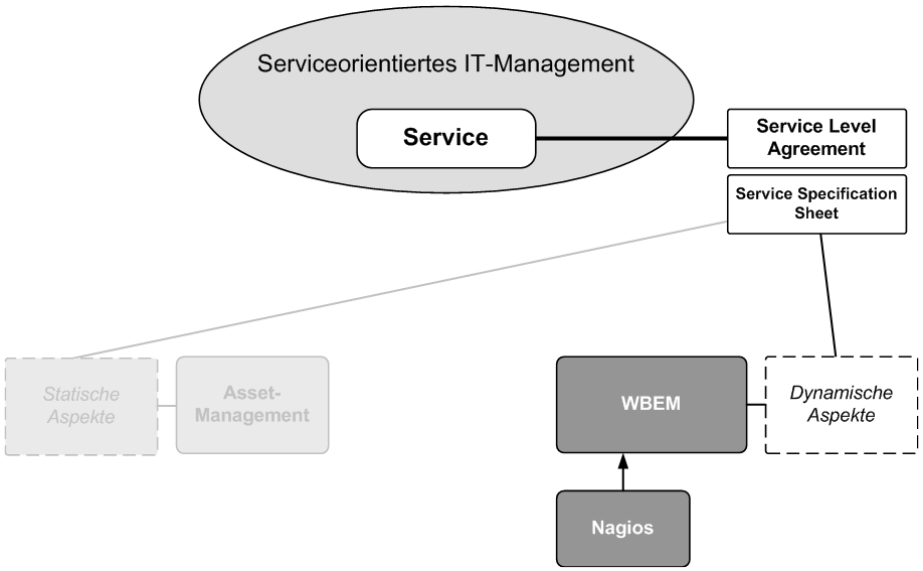

Abbildung 41: Nagios als WBEM-Zulieferer

Es stellt sich die Frage, zu welchem Zeitpunkt bzw. an welchen Stellen im Verarbeitungsprozess von Nagios eine Datenübergabe überhaupt stattfinden kann. Hierbei ergeben sich prinzipiell zwei verschiedene Varianten (siehe Abbildung 42). Diese werden im Folgenden in den Ansätzen 3 und 4 untersucht.

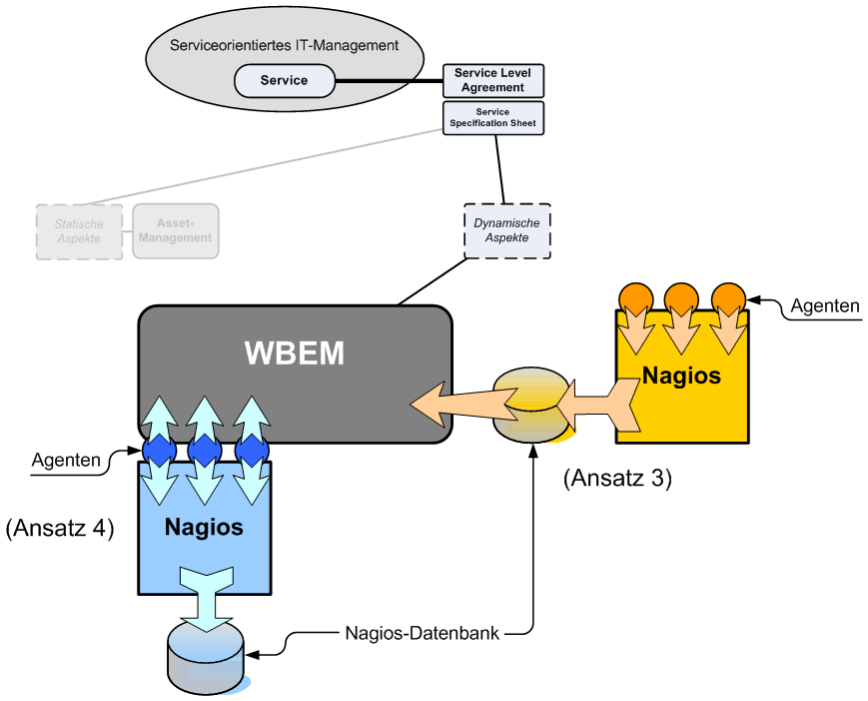

Abbildung 42: Ansätze 3 und 4

7.4 Ansatz 3 – Kopplung der Nagios-Datenbank an WBEM

Eine Variante ist, die für das WBEM-System interessanten Daten aus der Nagios-Datenbank auszulesen. Ein zu entwickelnder CIM-Provider greift stellvertretend für den CIM-Server direkt auf Felder der Nagios-Datenbank zu. Ausgelesene Daten werden im WBEM-Repository abgelegt und stehen dann für die weitere Verarbeitung zur Verfügung.
Für diese Vorgehensweise ist es erforderlich, dass der CIM-Provider fähig ist auf die SQL-Datenbank von Nagios zuzugreifen. Dem Provider muss bekannt sein, an welcher Stelle das abzugreifende Datum liegt. Dazu muss die Struktur des zugrunde liegenden NDOUtils-Datenmodells zu untersucht werden.

7.4.1 Bewertung von Ansatz 3

Vorteile

Der wichtigste Vorteil dieser Variante im Vergleich zu Ansatz 1 und 2 ist die Integration des WBEM-Standards. Hierdurch wird ein Schritt hin zu einem serviceorientierten Management durch eine offene WBEM-Architektur gemacht.

Nachteile

Die Datenstruktur von Nagios ist proprietär und nicht an einem Standardmodell wie z.B. CIM orientiert. Sie ist mit der Zeit parallel zur Weiterentwicklung der Software gewachsen und unterliegt somit auch weiterhin Änderungen, sobald sich an den Funktionen und Eigenschaften von neuen Nagios-Versionen Veränderungen ergeben. Somit sind permanente Anpassungsarbeiten am Mapping nötig, das beide Modelle miteinander verknüpft.
Nagios bleibt bei diesem Ansatz auch in Zukunft ein integraler Bestandteil der Management-Architektur. Da die Daten durch Nagios vorverarbeitet werden müssen, um in der angeschlossenen Datenbank zur Verfügung zu stehen, kann nicht einfach darauf verzichtet werden Nagios weiterhin einzusetzen. Dies wird dann zum Problem, aufgrund Nagios wegen unterschiedlicher Ursachen nicht mehr weiter entwickelt oder gepflegt wird oder sich andere Gründe ergeben, weshalb es nicht mehr eingesetzt werden soll.

7.5 Ansatz 4 – Integration von Nagios-Agenten in ein WBEM-System

Der vierte Ansatz beschäftigt sich damit, die Statusdaten bereits vor der Verarbeitung durch den Nagios-Hauptprozess abzugreifen. Dazu muss eine Möglichkeit gefunden werden, die Nagios-Agenten an das WBEM-System anzukoppeln. Die Statusdaten der überwachten Systeme fließen damit in zwei verschiedene Richtungen: Zum einen wie gewohnt für die Überwachung der Infrastruktur zu Nagios und zum anderen in das WBEM-System, wo sie weiteren Managementzwecken dienen.
Diese Methode folgt dem Grundgedanken des verteilten Managements: Die Managed Objects erzeugen Managementinformationen und halten sie vor; Manager holen sich bei Bedarf die Informationen ab uns verknüpfen sie miteinander.
Zur Realisierung dieses Ansatzes bieten sich ganz offensichtlich die schon verteilt installierten Nagios-Agenten (NRPE, NSCA) mit den zugehörigen Check Plugins an. Die Agenten liefern, sofern sie korrekt angesprochen werden, ihre Statusdaten nicht nur an einen Nagios-Prozess aus, sondern an jede beliebige anfragende Gegenstelle. Es muss auch hier ein CIM-Provider entworfen werden, der in der Lage ist, die Agenten auf korrekte Weise anzusprechen.

Die DMTF bietet mit den so genannten *Management Profiles* Spezifikationen an, die für ein bestimmtes Managementumfeld das CIM-Modell und das damit verbundene Verhalten definieren. Das Modell enthält die benötigten CIM-Klassen, Assoziationen, Ereignismeldungen (*indications*), Methoden und Eigenschaften.
Für die Verbindung der Nagios-Agenten an ein WBEM-System bietet das *Sensors Profile* großes Potential. Das Klassendiagramm ist in Abbildung 43 zu sehen.

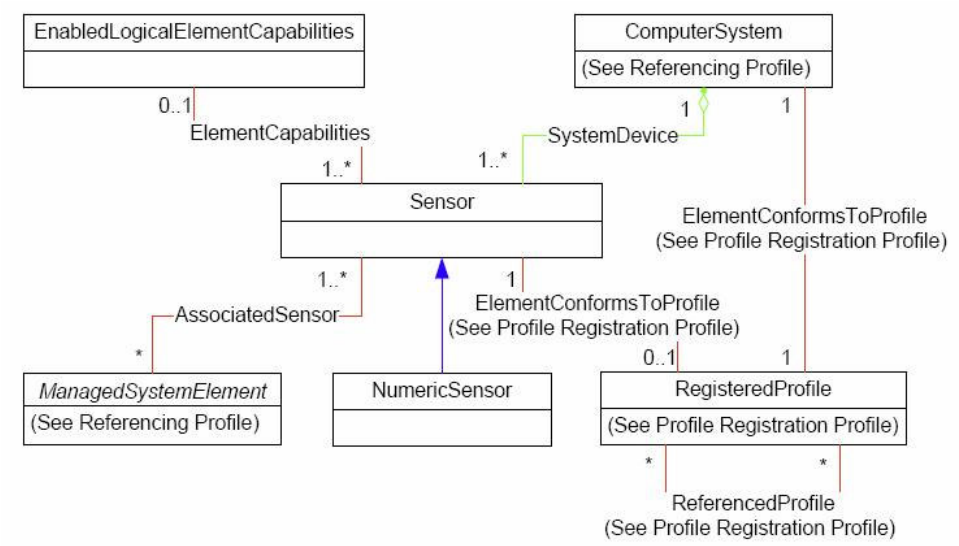

Abbildung 43: DMTF Sensors Profile Klassendiagramm [DMTF06a]

Die Eigenschaften der Klasse *Sensor* entsprechen in großem Maße denen des Objekttyps Host in Nagios. Das Attribut *CIM_Sensor.PossibleState* wird je nach Typ des Sensors mit drei potentiellen Statuswerten der Kategorien GOOD, BAD und UNKNOWN belegt. Dies entspricht weitgehend den möglichen Zuständen eines Nagios-Hosts (siehe 5.1.3).
Ebenfalls sehr interessant ist der in der Abbildung unten mittig befindliche *CIM_NumericSensor*. Er wird von der Klasse Sensor vererbt, enthält aber noch einige zusätzliche Attribute, die in Tabelle 6 aufgelistet sind.

Tabelle 6: Eigenschaften der Klasse CIM_NumericSensor

Properties and Methods	Requirement	Notes
SystemCreationClassName	Mandatory	Key
SystemName	Mandatory	Key
CreationClassName	Mandatory	Key
DeviceID	Mandatory	Key
BaseUnits	Mandatory	None
UnitModifier	Mandatory	None
RateUnits	Mandatory	None
CurrentReading	Mandatory	None
LowerThresholdNonCritical	Conditional	See section 7.5.
UpperThresholdNonCritical	Conditional	See section 7.6.
LowerThresholdCritical	Conditional	See section 7.7.
UpperThresholdCritical	Conditional	See section 7.8.
LowerThresholdFatal	Conditional	See section 7.9.
UpperThresholdFatal	Conditional	See section 7.10.
SupportedThresholds	Mandatory	See section 7.11.
SettableThresholds	Mandatory	See section 7.12.
SensorType	Mandatory	None
PossibleStates	Mandatory	See section 7.3.
CurrentState	Mandatory	See section 7.4.
ElementName	Mandatory	See section 7.19.
OtherSensorTypeDescription	Conditional	See section 7.17.
EnabledState	Mandatory	See section 7.16.
RequestedState	Mandatory	See section 7.14.
OperationalStatus	Mandatory	None
HealthState	Mandatory	None
RequestStateChange( )	Conditional	See section 8.1.
RestoreDefaultThresholds( )	Conditional	See section 8.2.

CIM_NumericSensor enthält mehrere Schwellwertdefinitionen. Ein kontinuierlich (analog) gemessener Parameter wird dann mit diesen Schwellwerten verglichen und bestimmt den aktuellen Zustand des Attributs *CIM_NumericSensor.CurrentState*. Dieses kann deutlich mehr Zustände annehmen als das entsprechende *CIM_Sensor.PossibleState-Attribut*. Es ist zu erkennen, dass der *CIM_NumericSensor* dem Objekttyp *Service* in Nagios sehr ähnlich ist.
Zusammenfassend ergibt sich damit das Potential, die Nagios-Entities auf einfache Weise vollständig auch in CIM abzubilden.

Bei genauerer Betrachtung lässt sich das Problem erkennen, dass der NRPE üblicherweise ‚nur' lokale Service Checks ausführt. Auf Prüfungen, die der Nagios-Hauptprozess direkt ausführt, wie *host_checks* und Checks von Netzdiensten hat der vorgeschlagene CIM-Provider damit auf den ersten Blick keinen Zugriff, da er seine Informationen ausschließlich über den Nagios-Agenten bezieht. Dies lässt sich jedoch durch einen einfachen Mechanismus beheben, wie er in Abbildung 44 gezeigt wird: Der NRPE ist in der Lage ist, beliebige Checks, also nicht nur Service Checks auszuführen. Wird nun ein NRPE-Proxy-Host eingerichtet auf dem die erforderlichen Plugins für *host_checks* und Checks von Netzdiensten installiert sind, so besteht für einen CIM-Provider die Möglichkeit, dort die fehlenden checks durchzuführen.

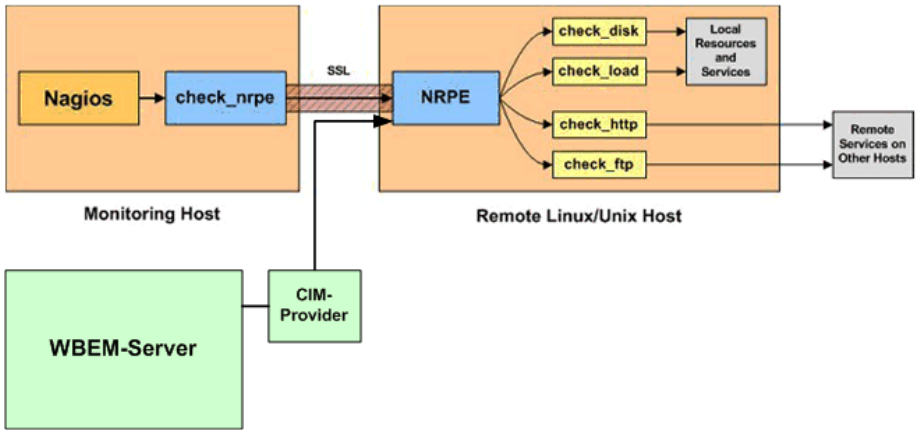

Abbildung 44: Check von lokalen und remote Services [Gal07d]

7.5.1 Bewertung von Ansatz 4

Vorteile

Neben den bei Ansatz 3 aufgeführten Vorteilen, die auch hier in vollem Umfang gelten, gibt es hier noch weitere:
Die Ermittlung von Statusdaten ist vollkommen unabhängig vom Nagios-Hauptprozess. Falls auf irgendeinem Grund später auf Nagios als Monitoring-Tool verzichtet werden soll, so ist dies im Gegensatz zu Ansatz 3 ohne Einschränkung des WBEM-Systems möglich. Lediglich die Agenten und die lokal installierten Nagios-Plugins müssen vorhanden sein und weiterhin gewartet werden.
Da die Daten direkt am Host abgeholt werden und weniger Komponenten am Fluss der Daten beteiligt sind, ist dieses Prinzip weniger anfällig für Ausfälle.

Nachteile

Der Aufwand zur Abfrage der Daten ist gegebenenfalls etwas höher im Vergleich zum Abrufen der von Nagios vorverarbeiteten Daten (Ansatz 3).

7.5.2 Weiterführung der Ansätze 3 und 4

Insgesamt erkennt man, dass die Ansätze 3 und 4 durch Ihre Verbindung mit einer WBEM-Architektur, das größte Potential in Bezug auf ein integriertes, serviceorientiertes IT-Management bieten. Die vorgeschlagenen Konzepte zur Integration des Monitoring-Systems Nagios in eine WBEM-Architektur noch detaillierter auszuführen und prototypisch umzusetzen übersteigt den Rahmen der vorliegenden Studienarbeit und kann gegebenenfalls in nachfolgenden Arbeiten aufgegriffen werden.

8 ZUSAMMENFASSUNG UND AUSBLICK

Nachfolgend werden die Ergebnisse dieser Studienarbeit noch einmal zusammengefasst. Für die beiden Hauptbestandteile dieser Arbeit, NSM und Nagios, wird das Erreichte bezüglich der Ausrichtung auf IT-Management und Serviceorientierung dargestellt. Abschließend werden Anknüpfungspunkte für weitergehende Arbeit gezeigt.

8.1 Zusammenfassung

Es werden zur Einführung das Themengebiet und grundlegende Begriffe beschrieben. Insbesondere wird genauer auf die Interpretation von Begriffen wie *IT-Service* eingegangen, deren Bedeutung sich abhängig vom Kontext, in dem sie benutzt werden, unterscheiden können.
Vom IT-Management ausgehend wird auf IT-Service-Management und die dazu gehörenden Begriffe eingegangen. Damit wird vorgegeben, welche Zielsetzungen für modernes IT-Management verfolgt werden. Durch sie ergeben sich die Anforderungen, denen sich Anwendungen wie die hier beschriebenen NSM und Nagios in Zukunft stellen müssen, sollen sie im Rahmen eines integrierten Managements genutzt werden.
Es wird gesondert auf Teilbereiche des IT-Managements wie die CMDB, *Service Specification Sheets*, WBEM und CIM eingegangen. Da diese zentrale Bestandteile dieser Arbeit sind, werden sie ausführlich dargestellt.

8.1.1 Steinmayr NSM

Eine Einführung in die Software NSM der Fa. Steinmayr wird gegeben. Es werden Aufbau, im Rahmen dieser Arbeit relevante Funktionalität und die Systemumgebung beschrieben, in der die Software normalerweise läuft. Die in NSM genutzten Begriffe werden geklärt und ein grundlegender Workflow für die Erfassung einer Infrastruktur beschrieben. Damit wird ein hinreichend genauer Einblick in die Software gegeben, um die Betrachtung bzgl. des IT-Managements zu ermöglichen. Die Umsetzung des hier genutzten Bespielszenarios in Cable-NSM wird beschrieben.
Darauf folgend wird das von Cable- und Logic-NSM genutzte Datenmodell der dahinter stehenden Datenbank beschrieben. Anhand einer tabellarischen Auflistung der Tabellen und Spaltentypen wird eine grobe Rekonstruktion des Datenmodells vorgenommen.
Es wird dargestellt, wie ein Ansatz aussehen kann, Aspekte moderner, serviceorientierter Betrachtungsweisen für eine Infrastruktur mit den bereits in NSM enthaltenen Informationen zu kombinieren.
Es wird das *Service Specification Sheet* nach ITIL so interpretiert, dass eine Implementation in NSM möglich wird und so eine serviceorientierte Sichtweise in NSM einführt. Durch diese wird eine neue hierarchische Struktur in NSM aufgenommen, der *Service-Baum*, der eine im Kontext von NSM sinnvolle Darstellung von IT-Services ermöglicht.
Anschließend wird eine prototypische Umsetzung vorgenommen. Die Fa. Steinmayr nimmt hierfür die für notwendigen Anpassungen an der Software vor; das Beispielszenario dient als Grundlage für eine Einschätzung, welchen Nutzen die Erweiterung von NSM im Rahmen des IT-Service Managements mitbringt.

8.1.2 Nagios

Zunächst wird der Aufbau der Monitoringsoftware Nagios beschrieben und genauer auf die im weiteren Verlauf relevanten Komponenten eingegangen. Die zentralen Nagios-Objekttypen *Host* und *Service* werden erklärt und in einem Informationsmodell in Relation zueinander gesetzt. Es wird die wichtige Rolle der Checks bzw. Plugins und deren Funktionsweise beschrieben. Die Manager-Agent-Architektur von Nagios wird anhand von NRPE und NSCA genauer untersucht. Schließlich wird ein kurzer Überblick über den derzeitigen Einsatz von Nagios in der ATIS gegeben.

Die Rolle des *Service Specification Sheets* für ein Monitoringsystem wird untersucht. Anhand des bei der statischen Servicemodellierung entwickelten Konzepts des Servicebaums werden die für das Monitoring eines IT-Service relevanten Objekte identifiziert.
In vier unterschiedlichen Ansätzen wird untersucht, wie mittels der Software Nagios die für den IT-Service relevanten Statusdaten erfasst und dargestellt werden können. Es werden Möglichkeiten aufgezeigt, wie Nagios in eine WBEM-Architektur für serviceorientiertes IT-Management integriert.

8.1.3 Konzeptzusammenfassung

Die Betrachtungen von NSM und Nagios ergeben, dass zwischen den in der jeweiligen Managementdomäne betrachteten Objekten Gemeinsamkeiten bestehen. In beiden wird eine Ausprägung eines Servicebaums, modelliert nach einem *Service Specification Sheet*, benutzt, um eine Repräsentation des Services in der Software zu entwickeln. Die Ausprägungen unterscheiden sich nach statischen Aspekten, d.h. Informationen, die erfasst werden können, ohne dass der Service erbracht wird, und dynamischen Aspekten, d.h. nach Daten, die zur Laufzeit erhoben werden. Beiden gemein ist ein Service-Objekt als Kernelement, das mit Objekten der bestehenden Datenbasis beider Programme verknüpft werden kann.
Damit zeigt sich, dass der Servicebaum ein geeignetes Hilfsmittel für eine gemeinsame Grundlage für ein serviceorientiertes Management zumindest in den Anwendungsbereichen von NSM und Nagios sein kann.

8.2 Ausblick

Serviceorientierung im IT-Management stellt Anforderungen an die für das Management genutzte Software. Aspekte, die bisher vor allem im Kontext ihrer jeweiligen Management-Domäne betrachtet wurden, müssen für eine Entwicklung hin zu einer am Service orientierten Betrachtungsweise in anderen, vorher nicht beachteten, Kontexten nutzbar gemacht werden.
Soll sich Software für das IT-Management in eine Architektur integrieren können, ist der einzig gangbare Weg hierfür über offene Standards; CIM/WBEM-basierte Architekturen können die Grundlage bilden für eine fortschreitende Entwicklung.
Der in dieser Arbeit im Mittelpunkt stehende Servicebaum könnte so auch eine wichtige Rolle in einer serviceorientierten Management-Architektur spielen. Verschiedene Ansätze sind denkbar (Abbildung 45):

- Die Modellierung eines Servicebaums in einem unabhängigen Teil der Management-Architektur; *Asset-/Configuration-* und *Fault-/Performance-Management* übernehmen liefern Daten (1)

- Die Modellierung eines Servicebaums in den Management-Applikationen selbst; dabei beschränken sie sich auf die jeweils notwendigen Aspekte. Datenaustausch findet über standardisierte Schnittstellen statt (2)

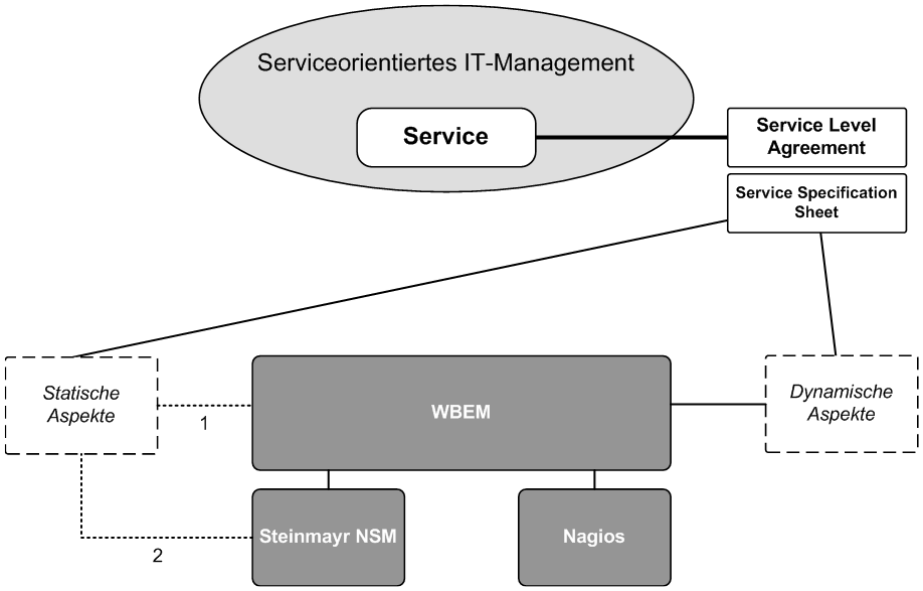

Abbildung 45: WBEM-Integration

8.2.1 Eine WBEM-Architektur als Bindeglied zwischen Management-Applikationen

In jedem Fall würden sich die Anwendungen selbst in den Kontext einer Management-Architektur auf Basis offener Standards stellen. Sie würden so weiterhin ihre Kernfunktion erfüllen, aber ihre Daten einer übergeordneten Architektur zur Verfügung stellen und gegebenenfalls an anderer Stelle gewonnene Informationen nutzen können.
Für NSM bedeutet dies, dass eine Anbindung an WBEM entwickelt werden müsste. So könnte das in Kapitel 4.4 beschriebene Service-NSM selbst zum CIM-Provider werden. Eine Unterscheidung von Service-NSM für die Erfassung statischer Aspekte eines IT-Service und Sensor-NSM (Kapitel 4.5) für das Monitoring ließe dieses in einer CIM-Architektur zu CIM-Sensoren werden. Wie eine Anbindung für Nagios an ein WBEM-System realisiert werden kann, wird in Kapitel 7 gezeigt.
Weitergehende Arbeit könnte zeigen, wie die in 8.1.3 genannten gemeinsamen Aspekte eines Servicebaums sich so durch die Nutzung von offenen Standards zusammenführen ließen.
Daraus ergäben sich Vorteile für das serviceorientierte IT-Management:

- Es ließe sich ein Servicebegriff im WBEM-System bilden, für den Daten aus *Asset-/Configuration-* und *Performance-Management* genutzt werden könnten
- Würde ein gangbarer Weg gefunden werden, die so in einen Kontext gestellten Objekte zu verknüpfen, könnten Informationen für andere Teile der Architektur nutzbar gemacht werden.

Damit wird auf Basis der offenen Standards WBEM und CIM serviceorientiertes IT-Managment unterstützt.

9 ANHÄNGE

9.1 Abbildungsverzeichnis

9.2 Index

9.3 Abkürzungen und Glossar

Abkürzung oder Begriff	**Langbezeichnung und/oder Begriffserklärung**
ATIS	*Abteilung Technische Infrastruktur*, eine Abteilung der Fakultät für Informatik der Universität Karlsruhe. Sie ist fakultätsintern verantwortlich für den Betrieb, die Instandhaltung und der Ausbau der Datennetzinfrastruktur und darauf basierender IT-Dienste.
Bautyp	Siehe: *Komponenten-Typ*
Cable-NSM	Zentrales Modul der Software Steinmayr NSM. Zuständig für Planung, Darstellung und Verwaltung von IT- und TK-Infrastrukturkomponenten
CIM	*Common Information Model*
CMDB	*Configuration Management Database*.
E-Mail-Dienst	IT-Dienst, der andere Teildienste wie *Mail empfangen* und *E-Mail senden* umfasst; hier: Kern-Maildienst der ATIS
Host (Nagios)	Ein von Nagios überwachtes (üblicherweise physikalisch greifbares) Objekt, das über eine Adresse erreichbar ist.
Host-Gruppe (Nagios)	Objekttyp zur Zusammenfassung von mehreren (funktional zusammengehörigen) Hosts
ITIL	Information Technology Infrastructure Library; umfassende und öffentlich verfügbare fachliche Dokumentation zur Planung, Erbringung und Unterstützung von IT-Serviceleistungen [ITIL08]. Der de-facto-Standard für die Umsetzung von *IT-Service-Management*
IT-Infrastruktur	Organisation und deren Bestandteile, die Anbieter von IT-Diensten für Dienstkunden ist
IT-Service	Ein Bündel von Nutzeffekten, das

	• durch Aktivitäten eines *Service Providers* erbracht wird, • durch IT- und Nicht-IT-Einrichtungen erzeugt wird, • vom *Service-Provider* an Servicekunden verkauft wird, • den Mitarbeitern des Servicekunden sowie anderen berechtigten Personen (Servicenutzern) bereitgestellt wird, • von den Servicenutzern eingesetzt wird, um ihre geschäftlichen Aufgaben auszuführen bzw. zu unterstützen.
IT-Service-Management	Oberbegriff für diejenigen Aspekte des IT-Managements, die zu einer bestmöglichen Unterstützung des Kunden eines IT-Service beitragen
Komponente	Wichtiger Begriff in Cable-NSM. Alle Objekte im *Komponentenbaum* sind vom Typ *Komponente*
Komponentenbaum	Hierarchische Datenstruktur in *Cable-NSM*. Enthält alle Anlagegüter einer in Cable-NSM erfassten Infrastruktur.
Logic-NSM	Modul von NSM. Es dient der Erfassung von aktiven Infrastrukturkomponenten und dem Abgleich des Datenbestands von *Cable-NSM* mit dem wirklichen Bestand
Nagios	Eine Monitoringsoftware zum Überwachen von Hosts und Services im Netzwerk.
NagVis	Addon für Nagios, das die Erstellung von Karten erlaubt
Nutzung	Eine *Nutzung* gibt im Rahmen von NSM einer Leitung eine Bezeichnung
OLA	*Operations Level Agreement*. Vereinbarung zwischen dem Erbringer einer Leistung und dem Abnehmer, wobei beide Teil der gleichen Organisation sind
Service (Nagios)	Ein von Nagios überwachtes Objekt, das auf einem Host läuft.
Service-Gruppe (Nagios)	Objekttyp zur Zusammenfassung von mehreren (funktional zusammengehörigen) Services
Service Specification Sheet	Form einer Dokumentation des *Service-Providers* darüber, wie Ressourcen seiner Infrastruktur zur Erbringung eines Service beitragen
UC	*Underpinning Contract*. Vereinbarung zwischen dem Erbringer einer Leistung und dem Abnehmer, wobei beide unterschiedlichen Organisationen angehören
Verbindung	Bezeichnet eine Leitung zwischen zwei Anschlüssen in Cable-NSM
WBEM	Web-Based Enterprise Management, ein Satz von Management- und Internet- Standardtechnologien zur Vereinheitlichung von

Literatur

[BMC06] BMC Software, *Was muss eine CMDB leisten?*, Grundsatzdokument, 2006

[CN04] Viktor Clerc, Frank Niessink, *IT Service CMM – a pocked guide*, Van Haren Publishing, 2004

[DMTF06a] Distributed Management Task Force, Inc. (DMTF), *DMTF Tutorial*, 2006 http://www.wbemsolutions.com/tutorials/DMTF/dmtftutorial.pdf

[DMTF06b] Distributed Management Task Force, Inc. (DMTF), *Computer System Profile*, Version 1.0.0b, 2006 http://www.dmtf.org/standards/published_documents/DSP1052.pdf

[DMTF06c] Distributed Management Task Force, Inc. (DMTF), *Sensors Profile*, Version 1.0.0c, 2006 http://www.dmtf.org/standards/published_documents/DSP1009.pdf

[DMTF07] DMTF SysDev-wg, *Core Specification V2.16*, 2007 http://www.dmtf.org/standards/cim/cim_schema_v2171/CIM_Core.pdf

[DW06] Distributed Management Task Force Inc. und WBEM Solutions, Inc., *DMTF Tutorial*, 2006 http://www.wbemsolutions.com/tutorials/DMTF/dmtftutorial.pdf

[Gal07a] Ethan Galstad, *Nagios Version 3.x Documentation*, 2007 http://nagios.sourceforge.net/docs/nagios-3.pdf

[Gal07b] Ethan Galstad, *NDOUTILS Documentation Version 1.4*, 2007 http://nagios.sourceforge.net/docs/ndoutils/NDOUtils.pdf

[Gal07c] Ethan Galstad, *NDOUtils Database Model*, 2007 http://nagios.sourceforge.net/docs/ndoutils/NDOUtils_DB_Model.pdf

[Gal07d] Ethan Galstad, *NRPE Documentation*, 2007 http://nagios.sourceforge.net/docs/nrpe/NRPE.pdf

[Heid03] Emanuel Heidinger, *Ein CIM-basiertes Modell der Rechnernetzepraktikum-Infrastruktur*, Institut für Informatik der TU München, 2003

[Hob04] Chris Hobbs, *A Practical Approach to Wbem/CIM Management*, Auerbach Publications, 2004

[Hup06] Paul G. Huppertz, *ITSM Advanced Pocket Book, Band 6: IT-Service – Der Kern des Ganzen*, liveIT!L, 2006

[ITIL08] *Homepage der ITIL*, 6.5.2008,http://www.itil.org/de/

[ITIL] IT Information Library Service Delivery, Service Support, ICTIM

[ITU97] ITU-T, *M.3400: TMN management functions*, 1997

[ITP07] IT Process Wiki, *Checklist Service Specification Sheet*, Version vom 02.11.2007

[Pan07] Ingo Pansa, *Evaluierung eines Nachfolgesystems für das Netzwerk- und System-Monitoring in der ATIS*, Studienarbeit, Universität Karlsruhe (TH), Institut für Telematik C&M, 2006

[Pan08] Ingo Pansa, *Erfassung von Abhängigkeiten zwischen IT-Ressourcen in*

einem Service-Kontext, Diplomarbeit, Universität Karlsruhe (TH), Institut für Telematik C&M, 2007

[Rie06] Götz Rieger, *Netzwerk unter Kontrolle*, Heise Zeitschriften Verlag, 2006 http://www.heise.de/netze/artikel/81238/

[Ste06a] Steinmayr Net Intelligence GmbH, *Schulung Logic-NSM*, Schulungsdokument, 2006

[Ste06b] Steinmayr Net Intelligence GmbH, *Praxisbeispiel Compact-NSM*, 2006

[Ste08] Steinmayr Net Intelligence GmbH, *Netz-Service-Management (NSM)*, 2008 http://www.steinmayr.de/Steinmayr/

[TSO01] *Best Practise for Service Delivery*, The Stationery Office, 2001