Universität Leipzig

Fakultät für Mathematik und Informatik

Institut für Informatik

Sicherheitsmanagement für Dateisysteme in Windows NT-Netzwerken per Tivoli TME 10 – Entwicklung eines Modellierungstools für die Nutzerverwaltung

Diplomarbeit

Leipzig, Oktober 2002

vorgelegt von
Bert Stallbaum

meinen Eltern

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Viele Unternehmen setzen zur Bewältigung ihrer Aufgaben DV-Anlagen ein. Dabei handelt es sich häufig um eine Vielzahl von Computern und ergänzender Peripheriegeräte. Je nach Einsatzzweck dienen diese Computer in einem Unternehmen als Arbeitsplatzrechner, Abteilungsserver oder zentrale Server. Dementsprechend unterscheiden sich die Anforderungen an die Leistungsfähigkeit dieser Maschinen, woraus der Einsatz unterschiedlicher Hard- und Software resultiert.
Die Erstellung einer maßgeschneiderten Konfiguration der Zugangsmöglichkeiten eines Benutzers zur DV-Technik eines Unternehmens (auch Behörden etc.) stellt eine Herausforderung an die jeweiligen Systemverantwortlichen dar. Dabei müssen die Nutzer auf Programme und Daten, die gegebenenfalls verteilt auf vernetzten Computern gespeichert sind, teils lesend, teils verändernd zugreifen. Der Zugriff der Nutzer auf DV-Ressourcen sollte so limitiert bzw. kontrolliert erfolgen, dass

1.) allgemeine Prinzipien des Datenschutzes umgesetzt werden, d.h. die Nutzer erhalten nur Zugang zu Daten, die sie tatsächlich für ihre Arbeit benötigen und
2.) die Möglichkeit, ungewollt, versehentlich Veränderungen an der Konfiguration von Programmen oder Datenbeständen zu verursachen, a priori eingeschränkt wird.

Durch die Systemverantwortlichen werden deshalb Berechtigungen vergeben, mit denen die Benutzer auf Ressourcen zugreifen dürfen. Die Vielzahl der im Unternehmen eingesetzten Applikationen, mit jeweils eigenen Konfigurationsmöglichkeiten und -anforderungen, trägt zur Komplexität der Problematik bei. Die Berechtigungen zum Zugriff auf Ressourcen werden im Allgemeinen als Zugriffsberechtigungen bezeichnet. Einige Betriebssysteme bieten – mittels enthaltener Werkzeuge und interner Strukturen – Ansätze zur Erteilung und Verwaltung dieser Zugriffsberechtigungen, die sich jedoch in Konzept und Ausprägung unterscheiden. In einer heterogenen Umgebung müssen daher unterschiedliche Sicherheitsmechanismen gemanagt werden. Dabei ist der Einsatz von Systemmanagementwerkzeugen unerlässlich.

Hinweis:
Warennamen und Markennamen sind in dieser Arbeit nicht besonders gekennzeichnet und werden ohne Gewährleistung der freien Verfügbarkeit benutzt. Sie sind möglicherweise eingetragene Warenzeichen.

1.1 Motivation und Problemstellung

Die *R+V Versicherung*, ein Versicherungsunternehmen mit Sitz in Wiesbaden, nutzt zur Datenverarbeitung eine umfangreiche IT-Landschaft. Der Bereich *Zentrale Informationssysteme* in der Direktion in Wiesbaden betreut sowohl die DV-Technik an den zentralen Standorten in Wiesbaden als auch in den zahlreichen bundesweit verteilten Filialen. Dabei müssen Server und Endgeräte administriert werden. Zur Verwaltung der firmeneigenen IT sollen u.a. Produkte aus der Systemmanagement-Softwaresuite „TME 10“ des Herstellers „Tivoli Systems“ verwendet werden. Für die Server – u.a. sind die Betriebssysteme MVS, UNIX und Windows NT 4.0 im Einsatz – ist die Einrichtung eines zentralen Managements für Zugriffsrechte und Benutzer geplant, um u.a. Probleme der folgenden Art zu adressieren:

- Über welche Zugriffsberechtigungen verfügt ein Benutzer insgesamt? Derzeit lässt sich diese Frage (mit Hilfe der herkömmlichen Mittel der jeweiligen Systeme) nicht mit vertretbarem Aufwand beantworten.

- Welche Zugriffsberechtigungen benötigt ein bestimmter Nutzer tatsächlich und welche müssen ihm entzogen werden, weil sie beispielsweise nur für die Dauer eines mittlerweile abgeschlossenen Projektes erforderlich waren?

Um derartige Fragen beantworten zu können, wurden bei der R+V Versicherung zwei grundlegende Entscheidungen getroffen:

1. Die benötigten Informationen werden zentral in einer Datenbank gespeichert.

2. Die Administration der Zugriffsrechte soll mit Hilfe des Produktes „TME 10 Security Management“ erfolgen.

In dieser Arbeit wird „TME 10 Security Management“ auch als TSecMan abgekürzt. „TME 10 Security Management“ verspricht, eine zentralisierte Verwaltung von Zugriffsberechtigungen über unterschiedliche Betriebssysteme hinweg (darunter auch Windows NT 4.0) zu ermöglichen. Um dies zu erreichen, unterhält TSecMan ein eigenes rollenbasiertes Sicherheitsmodell, das auf die Sicherheitssysteme der verwalteten Betriebssysteme abgebildet wird. Zuvor muss dieses rollenbasierte Sicherheitsmodell, das im Weiteren auch synonym als Rollenmodell bezeichnet wird, allerdings entsprechend den individuellen Gegebenheiten des jeweiligen Unternehmens sowie den Wünschen und Forderungen der Systemverwalter erarbeitet werden. Anschließend kann es dann in TSecMan umgesetzt werden.

Wird TSecMan in eine bestehende EDV-Umgebung eingebracht, so kann im Idealfall auf ein bereits bestehendes Rollenmodell oder auf Ansätze zu selbigem zurückgegriffen werden. Für Fälle, in denen das Rollenmodell noch erarbeitet, formuliert oder verfeinert werden muss, kommt es u.a. darauf an, Informationen über die bestehenden System/Server-Konfigurationen zu erfassen und zu analysieren, um so eine Grundlage für das zu erstellende Sicherheitsmodell zu erhalten. In jedem Fall ist es unabdingbar, den aktuellen Zustand der eingerichteten Zugriffsberechtigungen zu ermitteln und zu dokumentieren.
Für die Windows NT 4.0 Netzwerke der R+V Versicherung fehlt bisher ein umfassendes Rollenmodell. Um erteilte Zugriffsberechtigungen eines Nutzers zu bestimmen, müssen – im Hinblick auf Zugriffsberechtigungen, die für Gruppen erteilt wurden – insbesondere auch die Gruppenmitgliedschaften des Nutzers betrachtet werden. Im Rahmen der Planung eines Rollenmodells ist es daher notwendig, die betreffenden (NT-)Ressourcen samt der erteilten Zugriffsberechtigungen sowie die vorhandenen Gruppen und Nutzer zu erfassen.
TSecMan bietet für die Erfassung von bestehenden Serverkonfigurationen teilweise halbautomatische Dienste. Allerdings ist die derzeit verfügbare Unterstützung für Windows NT in Bezug auf Ressourcen, insbesondere für Verzeichnisse, unzureichend. Der Administrator muss entsprechende Informationen manuell eingeben, was in größeren Umgebungen in einer für ihn sehr zeitaufwendigen Arbeit resultiert und nur mit erheblichem Aufwand durchführbar ist.
Seitens der R+V Versicherung wurde diese Problematik mit Vertretern des Instituts für Informatik der Universität Leipzig erörtert. Ein Ergebnis dieser Gespräche war der Vorschlag, eine entsprechende Softwarelösung zu entwickeln.
Aufgabe der vorliegenden Arbeit ist es daher:

1. die prinzipielle Funktionsweise von TSecMan im Hinblick auf ein zu integrierendes Rollenmodell zu untersuchen, aufzubereiten und darzustellen,
2. die Grundlagen für die Erfassung des Ist-Zustandes eines Windows NT 4.0 Netzwerkservers in Bezug auf vorhandene Gruppen, Nutzer und Ressourcen samt der erteilten Zugriffsberechtigungen herauszuarbeiten,
3. eine Softwarelösung für die Erfassung der benötigten Informationen zu Windows NT 4.0 Netzwerken zu entwerfen, zu programmieren und zu implementieren. Diese soll als Ausgangspunkt zur Entwicklung weiterer Administrationsanwendungen – für Reporting- und Modellierungszwecke bei der Erarbeitung eines Rollenmodells – dienen.

Die ermittelten Daten sind in einer relationalen Datenbank (Oracle) abzulegen. Die Anwendung ist als 3-stufige Client/Server Applikation unter Einsatz von Java und CORBA zu implementieren. Die Entscheidung für CORBA fiel im Hinblick auf die Einbindungsmöglichkeiten weiterer Plattformen der heterogenen

IT-Infrastruktur der R+V Versicherung. Als Entwicklungsumgebung für den CORBA-bezogenen Teil der zu erstellenden Software, wird „Visibroker für Java“ der Firma Borland verwendet. Die Anzeige soll auf dem Browser des Clients erfolgen, siehe Abbildung 1.

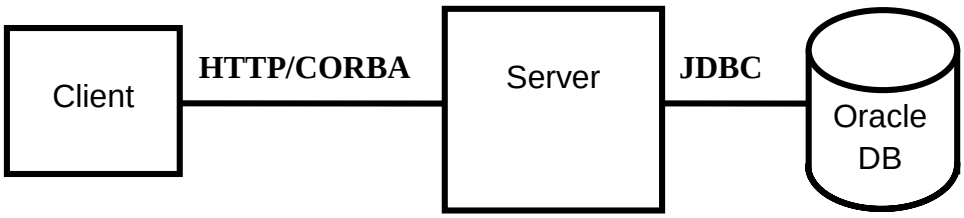

Abbildung 1: schematische Darstellung der vorgegenbenen Randbedingungen (zur Architektur) der angestrebten Softwarelösung

1.2 Gliederung der vorliegenden Arbeit

Im Anschluss an diese Gliederung beginnt Kapitel 2. Dieses Kapitel soll das Verständnis für die grundlegende Problematik des Netzwerk- und System-managements (NSM) heterogener IT-Umgebungen fördern und damit der Ein-ordnung des Themas der vorliegenden Arbeit dienen. Kapitel 2 soll insbesondere verdeutlichen

- was Sicherheitsmanagement (im Allgemeinen) bedeutet,
- welche Disziplinen u.a. dazugehören und
- dass Sicherheitsmanagement ein Teilbereich des NSM ist und als solcher auch im Zusammenhang mit anderen NSM-Bereichen betrachtet werden (kann und) sollte.

Das Sicherheitsmanagement ist selbst ein umfangreiches und komplexes Themengebiet. In der vorliegenden Arbeit steht das Teilgebiet „Zugriffskontrolle“ im Vordergrund, also der Teil des Sicherheitsmanagements, der sich mit der Verwaltung von Zugriffsberechtigungen auf (IT-)Ressourcen beschäftigt. Da diese Zugriffsberechtigungen für Benutzer gelten, ist der Bereich Zugriffskontrolle eng mit der „Benutzerverwaltung“ (ebenfalls ein Teilbereich des Sicherheitsmanagements) verflochten. Daher wird in der vorliegenden Arbeit auch auf Teilaspekte der Benutzerverwaltung eingegangen.
Kapitel 2 nimmt außerdem Stellung zu allgemeinen Aspekten und Problemen von NSM-(Software-)Werkzeugen (Managementsystemen). Eine Technik, die sich als Grundlage bzw. Bestandteil von NSM-Werkzeugen etabliert hat, verbirgt sich hinter der Abkürzung „SNMP“. Im letzten Teilabschnitt von Kapitel 2 wird auf einige

charakteristische Aspekte und auf Grenzen von SNMP eingegangen.
Kapitel 3 gibt einen Überblick zu CORBA. Hinter der Abkürzung bzw. dem Schlagwort „CORBA“ steckt eine Technologie, die insbesondere auch für das NSM heterogener IT-Landschaften interessant ist – wobei CORBA nicht auf das Anwendungsgebiet NSM beschränkt ist. Ein Grund, an dieser Stelle auf CORBA einzugehen, besteht darin, dass die Systemmanagement-Softwaresuite (TME 10) von Tivoli auf CORBA basiert. Kapitel 4 gibt einen Überblick zu Tivoli TME 10 und stellt dazu einige grundlegende Begriffe und Konzepte vor.
Kapitel 5 geht auf die Zugriffskontrolle für Verzeichnisse und Dateien innerhalb von Windows NT-Netzwerken ein und erläutert die dafür relevanten Sicherheitskonzepte (und Begriffe) von Windows NT 4.0.
Im Kapitel 6 wird die grundlegende Funktionsweise von TSecMan erläutert, wobei insbesondere die Aspekte/Details von TSecMan Gegenstand des Kapitels sind, die die Adminstration/Zugriffskontrolle in Windows NT-Netzwerken mittels TSecMan betreffen.
In Kapitel 7 wird das „Modellierungstool“, eine Softwarelösung für die Ermittlung von Informationen über vorhandene Windows NT-Umgebungen, vorgestellt.

2 Netzwerk- und Systemmanagement – ein Abriss

Dieses Kapitel soll zum Verständnis der grundlegenden Problematik des Netzwerk- und Systemmanagements beitragen und so der Charakterisierung des Fachgebietes dienen, in das die Aufgabenstellung der vorliegenden Arbeit einzuordnen ist. Es wird zunächst auf allgemeine Aspekte des Netzwerk- und Systemmanagements und anschließend auf ausgewählte Schwerpunkte eingegangen. Die hier gebotene Darstellung erhebt keinen Anspruch auf Vollständigkeit, sondern will lediglich einen Überblick liefern. Für weitergehende Betrachtungen soll an dieser Stelle auf die Literatur zu diesem Thema (insbesondere [Cyp92], [Fle98], [Heg99], [Jan93], [Sta93]) verwiesen werden.

2.1 Einführung

Die Begriffe „Systemmanagement“ (SM), „Netzwerkmanagement“ (NM), „integriertes Netzwerk- und Systemmanagement“, „Enterprise-IT-Management“ werden in der Literatur teils synonym, teilweise auch für unterschiedliche Bereiche verwendet. So konzentriert sich das Netzwerkmanagement hauptsächlich auf die Verwaltung von aktiven und passiven Elementen die zur Infrastruktur für die Datenübertragung gehören und in denen keine weitergehende Informationsverarbeitung stattfindet. Historisch gesehen ist das NM sehr stark durch die Entwicklung und den Einsatz des Managementprotokolls SNMP (Simple Network Management Protocol) geprägt, wobei anfangs die Überwachung und Fehlererkennung im Vordergrund stand. Auf dieses Protokoll und das zugehörige Konzept wird im Abschnitt 2.4 eingegangen. Für das Systemmanagement gibt es keine allgemein verbindliche Definition. Es befasst sich mit der Verwaltung eines verteilten Systems samt seiner Anwendungen. Das SM betraf zunächst die Verwaltung einzelner Rechner. Dabei ging es um Aufgaben, wie:

- Steuerung der Auslastung (Stapelverarbeitung),
- Benutzerverwaltung,
- Installation von Software,
- Datensicherung,
- Überwachung und Gewährleistung der Sicherheit des Systems.

Während anfangs Großrechner mit angeschlossenen Terminals dominierten, war die weitere Entwicklung der Rechentechnik durch die Verbreitung von Client/Server-Systemen geprägt. Im Gegensatz zu Terminals, fand nun auch auf der Seite der Clients Informationsverarbeitung statt. Damit wurde allerdings eine zunehmende

Abhängigkeit der Anwendungsprogramme von der Funktionstüchtigkeit des Netzwerkes eingeführt. Die Benutzbarkeit eines Anwendungsprogrammes ist gegebenenfalls auf die Erreichbarkeit und das Funktionieren mehrerer Server angewiesen. Im Fehlerfall kommen so zusätzliche mögliche Ursachen in Betracht. Für das Management der entstandenen IT-Landschaften ergaben sich neue Anforderungen. Daraus resultierte eine Konvergenz zwischen Netzwerk- und Systemmanagement, die eine klare Trennung beider Begriffe erschwert.
In dieser Arbeit soll das „Netzwerk- und Systemmanagement“ (NSM) für die Gesamtheit all der Einrichtungen und Tätigkeiten stehen, die erforderlich sind um eine IT-Landschaft funktionsfähig zu halten und anzupassen. Also alle Maßnahmen und Vorkehrungen, die dafür sorgen, dass die Mitarbeiter eines Unternehmens den EDV-gestützten Teil ihrer Arbeit erledigen können. Wenn es darum geht, große verteilte Systeme zu managen, treten mehrere Problemfelder des NSM deutlich zutage:

- Es müssen Produkte verschiedener Hersteller, unterschiedliche System-architekturen, lokale Netzwerke (LAN), Weitverkehrsnetze (WAN) sowie Systeme zur Daten- und Sprachübertragung verwaltet werden.
- Die Anforderung einer nahezu 100%igen Verfügbarkeit der Systeme wird immer wichtiger für das Funktionieren vieler Unternehmen.
- Die Systeme werden komplexer und umfassen mehrere Technologien und Terminologien.
- Das betroffene Personal für das NSM benötigt immer mehr Qualifikationen und Fähigkeiten. Schulungen sind zeitlich und finanziell kostenintensiv.
- Der Bedarf an Automatisierung für immer mehr Tätigkeiten wächst.
- Remote Monitoring (Fernüberwachung) für Teile des Netzes und Systemkom-ponenten ist erforderlich. Remote Control (Fernsteuerung) ist erwünscht.

Um dies bewältigen zu können, sind entsprechende Methoden, geeignete Werkzeuge und zuständiges, geschultes Personal (Administratoren) erforderlich. Das Spektrum der derzeit verfügbaren Werkzeuge umfasst u.a.:

- Skripte, vorgefertigt oder von Administratoren selbst erstellt,
- Werkzeuge, die zum Betriebssystem gehören,
- herstellerspezifische Werkzeuge zur Steuerung einzelner Hardwarekomponenten bzw. Produktfamilien,
- spezielle Managementsoftware, die über einen eingeschränkten Funktions- und Wirkungsbereich verfügt, aber einzelne Aufgaben des NSM löst,
- Management-Software-Suites, die bemüht sind, auf den weit verbreiteten Plattformen mehrere, bzw. möglichst alle, Managementgebiete – entweder

durch eigene Produkte oder durch Integration etablierter Produkte anderer Hersteller – abzudecken.

In die beiden letzten Kategorien sind insbesondere so genannte Mangementsysteme einzuordnen, die ein plattformübergreifendes Management vorhandener IT-Systeme bieten. Aus Sicht des Personals zur Systemverwaltung werden Mangementsysteme gewünscht, die eine zentrale Kontrolle der Netz- und Systemkomponenten ermöglichen. Wichtig ist, dass diese Werkzeuge an die individuellen Bedürfnisse der jeweiligen Systemverantwortlichen angepasst werden können, weil diese Personen die Anforderungen an die Verwaltung der EDV-Umgebung des Unternehmens am besten kennen. Da die eingesetzte IT-Landschaft für nahezu alle Unternehmen verschieden ist, gibt es für den Einsatz von Management-Software in Unternehmen einerseits individuelle Anforderungen und andererseits allgemeine Anforderungen, die individuell gewichtet werden müssen. Das führt unter anderem dazu, dass eine für ein Unternehmen A erarbeitete und angepasste Managementlösung im Allgemeinen nicht 1:1 für ein Unternehmen B übernommen werden kann. Anforderungen allgemeiner Natur sind beispielsweise:

- Gewährleistung einer konstant hohen Dienstgüte des Netzes (Quality of Service)
- flexible Anpassung des Netzes an veränderte Anforderungen
- Überwachung von Netzwerken, Applikationen und Servern, um Engpässe bzw. drohende Ausfälle zu erkennen und zu verhindern (Erhöhung der Verfügbarkeit)
- Abrechnung der Nutzung von IT-Ressourcen als Dienstleistung
- Erhöhung der Benutzerzufriedenheit, die aus der Perspektive des NSM hauptsächlich von der Performance und Betriebssicherheit von Applikationen abhängt
- effiziente Nutzung vorhandener IT-Ressourcen zur Erreichung eines optimalen Kosten-Nutzen-Verhältnisses
- Automatisierung von Routineaufgaben bei der Administration von Netzen und Systemen
- eingesetzte Werkzeuge sollen unter einer einheitlichen Bedieneroberfläche eine unternehmensweite Sicht und Steuerung der vorhandenen IT-Systeme bieten

2.2 Funktionsbereiche

Die ISO (International Organization for Standardization) hat auf dem Gebiet des NSM unter der Bezteichnung „OSI Systems Management“ (OSI: Open Systems Interconnection) mehrere Standards und Konzepte zu einem offenen, d.h. herstellerneutralen, Management vernetzter Systeme erarbeitet. Unter anderem wurden die generellen Aufgaben, die im Rahmen eines solchen Managements zu erbringen sind, untersucht. Dabei wurden folgende fünf Funktionsbereiche unterschieden [Lex98]:

- Fehlermanagement (fault management)
- Konfigurationsmanagement (configuration management)
- Abrechnungsmanagement (accounting management)
- Leistungsmanagement (performance management)
- Sicherheitsmanagement (security management)

Diese Einteilung stieß auch außerhalb des „OSI Systems Management“ auf breite Akzeptanz sowohl bei den Herstellern standardisierter als auch bei den Herstellern proprietärer Managementsysteme [Sta93]. Darüber hinaus existieren weitere Gliederungsmöglichkeiten für die Aufgabenbereiche des Managements. So wird etwa unter der Bezeichnung Bestandsmanagement das Konfigurationsmanagement um Funktionen ergänzt, die mit Bestellwesen zu tun haben [Heg99]. Das „Inventory Management“ fasst die mit der Dokumentation aller vorhandenen Komponenten verbundenen Aufgaben zusammen. Durch das „Asset Management“ wird es um eine betriebswirtschafliche Bewertung erweitert, wodurch Betrachtungen der Kosten einer IT-Infrastruktur („Cost of Ownership“) erleichtert werden. Letztendlich finden sich jedoch die obigen Funktionsbereiche auch in altenativen Gliederungsvorschlägen der Managementdisziplinen, teils in geändertem Umfang, wieder.

2.2.1 Fehlermanagement

Das Fehlermanagement beschäftigt sich mit der Erkennung, Isolation, Meldung und Korrektur von abnormalem Verhalten der verwalteten Komponenten. Das Ziel des Fehlermanagements besteht darin, die Verfügbarkeit des verwalteten verteilten Systems, also aller Hard- und Softwarekomponenten sowie der durch sie erbrachten Dienste, durch schnelle Entdeckung und Beseitigung von auftretenden Fehlern zu maximieren. Dabei werden Netz- und Systemzustände überwacht und Fehler-protokolle (Error Log) geführt. Fehler die nur durch Aktionen seitens des Managementsystems bzw. des Systempersonals korrigiert werden können,

bezeichnet man auch als „Faults“. Um Faults zu erkennen, kann beispielsweise die Häufigkeit „einfacher Fehler“ (z.B. Bitfehler bei der Datenübertragung) ausgewertet werden. Überschreitet die Häufigkeit eines Fehlers einen zuvor definierten Schwellwert, kann dies auf einen Fault hindeuten. Um die Ursache des Faults weiter eingrenzen zu können, ist meist die Auswertung weiterer Bedingungen, nach ebenfalls zuvor definierten Regeln, erforderlich. Dieser Vorgang wird als Korrelation bezeichnet. Auf Grund der vielfältigen Abhängigkeiten einzelner Komponenten in Rechnernetzen kann ein einzelner Fehler eine Vielzahl von Folgefehlern und zugehörigen Fehlermeldungen für das Management verursachen. Eine Schwierigkeit bei der Fehlerfindung besteht darin, aus der entstandenen Menge von Fehlermeldungen den Auslöser für die gemeldeten Folgefehler herauszufiltern. Idealerweise würde eine automatische Korrelation des Managementsystems die aufgetretenen Fehler auswerten, das Ergebnis als eine aussagekräftige Fehlermeldung an das Systempersonal senden und eventuell erste Maßnahmen zur Korrektur des Problems einleiten. In der Realität sieht es allerdings meist so aus, dass die Korrelation noch vom Systempersonal erbracht wird, das mit Hilfe von Diagnosetests die Problemursache isoliert. Lässt sich das Problem nicht sofort korrigieren, werden gegebenenfalls „Erste-Hilfe-Maßnahmen“ eingeleitet und ein Reparaturauftrag, z.B. als sog. „Trouble-Ticket“, an dafür spezialisiertes Servicepersonal gesendet. Ein „Trouble-Ticket-System“ verwaltet Fehlermeldungen als Dokumente. Dabei werden die einzelnen Schritte zur Bearbeitung des Problems von der Erfassung der Störung bis zur Fehlerbehebung dokumentiert, der aktuelle Bearbeitungsstand ist so jederzeit abrufbar. Wurde das Problem gelöst, sind die gegebenenfalls durchgeführten Erste-Hilfe-Maßnahmen zurückzunehmen. Nun gilt es, den Fehlerort noch eine Weile intensiv zu überwachen um sicherzustellen, dass die tatsächliche Fehlerursache behoben wurde. Der dafür zuständige Teil des Fehlermanagements wird als „Problem Tracking and Control“ bezeichnet.

2.2.2 Konfigurationsmanagement

Das Konfigurationsmanagement umfasst das in Betrieb nehmen und das außer Betrieb nehmen von Komponenten oder Subsystemen eines vernetzten Systems sowie das Anpassen dieser Komponenten an aktuelle Erfordernisse während des Betriebes. Dazu überwacht und manipuliert das Konfigurationsmanagement den Status verschiedener Parameter der verwalteten Komponenten. Dies kann z.B. als Reaktion auf festgestellte Fehler oder Lastveränderungen geschehen. Die Festlegung von Initialisierungswerten und Schwellwerten gehört ebenso zum Konfigurationsmanagement wie Software-Verteilung, Lizenzüberwachung und Datensicherung. Ein weiterer Schwerpunkt des Konfigurationsmanagements ist das Erfassen und Dokumentieren der vorhandenen Netz- und Systemkomponenten sowie ihrer

physischen und logischen Verbindungen. Die so gewonnenen Informationen können zur graphischen Darstellung des aktuellen Zustands der Komponenten und der Netzwerkstruktur nach geographischen oder logischen Aspekten sowie zur Generierung von Konfigurationsreports genutzt werden.

2.2.3 Abrechnungsmanagement

Beim Abrechnungsmanagement handelt es sich um die Erfassung von Kosten, die durch die Benutzung der vorhandenen IT-Systeme verursacht wurden, und ihre Zuordnung zu Nutzergruppen. Dabei ermöglicht das Abrechnungsmanagement die Definition und Überwachung von Kontingenten und Limits für die Nutzung der vorhandenen IT-Ressourcen. In der Praxis hängen die Ausprägung des Abrechnungsmanagements, der Umfang der zu erfassenden Daten und die Festlegung von Abrechnungstarifen sehr stark von firmenpolitischen Entscheidungen des betreffenden Unternehmens ab. Insbesondere ist auch der Aufwand der gewünschten Kostenerfassung mit deren Nutzen abzuwägen. Prinzipiell ermöglicht das Abrechnungsmanagement:

- die Erstellung von Rechnungen und deren Zustellung an Kostenstellen
- die Generierung von Reports und Statistiken über die entstandenen Kosten der IT
- das Aufdecken von ungewöhnlichen Netz- oder Systembelastungen durch exzessive, missbräuchliche Ressourcennutzung einzelner Benutzer
- zu Erkennen, ob Nutzer Schulungsmaßnahmen im Umgang mit den angebotenen Diensten benötigen (weil sie beispielsweise eine Software oder die Dienste eines Servers nicht oder falsch nutzen)
- die Gewinnung von Informationen für die Planung von Netz- und System-erweiterungen

2.2.4 Leistungsmanagement

Das Leistungsmanagement kann in Bezug auf seine Zielsetzung als Weiterführung des Fehlermanagements angesehen werden. Neben der Vermeidung von System-ausfällen geht es darum sicherzustellen, dass das zu verwaltende verteilte System hinsichtlich bestimmter Leistungskriterien optimiert läuft bzw. dass leistungsrelevante Parameter ein festzulegendes Mindestniveau nicht unterschreiten. Zu diesem Zweck werden Indikatoren und Kennzahlen zur Charakterisierung von Systemzuständen sowie der Güte von Diensten, die das verteilte System erbringt, definiert. Das Überwachen von Kenngrößen des verwalteten Systems wird als „Performance Monitoring“ bezeichnet, während man bei der Steuerung leistungsbestimmender Parameter von „Performance Controlling“ spricht. Das Leistungsmannagement überwacht die Zustände von Netzen, Rechnern, Betriebs-

systemen und Anwendungen. Dabei können beispielsweise Werte für

- Festplattenbelegung, Speicher und CPU-Auslastung von Rechnern
- Antwortzeiten von Anwendungen
- den Füllgrad des Tablespace einer Datenbank
- die Anzahl ausgeführter Transaktionen
- den Durchsatz eines Routers usw.

erfasst werden. In Dienstgütevereinbarungen (Service Level Agreements) können die zu erzielenden Leistungskriterien detailliert festgelegt werden. Neben Mittel- und Grenzwerten für angegebene Leistungsparameter können auch Messverfahren und Messstellen spezifiziert werden.

Eine besondere Schwierigkeit des Leistungsmanagements besteht darin, dass Kriterien, die (oder nach denen) es optimierend steuern soll, zum Teil nicht oder nur sehr schwer in messbare Größen gefasst werden können und teilweise nicht oder nur sehr schwer messbar sind. So ist das Antwortzeitverhalten von Anwendungen auf Seiten der Benutzerrechner von vielen Faktoren abhängig. Dazu zählen die Auslastung beteiligter Netze und Server sowie Belastungen des Benutzerrechners durch weitere Anwendungen. Für eine Beurteilung einzelner Leistungskriterien ist in verteilten Systemen häufig die Korrelation mehrere Indikatoren und Messdaten erforderlich. In der Praxis übersteigt der damit verbundene Aufwand leider in vielen Fällen die Grenze des Vertretbaren. Hinsichtlich der Antwortzeiten besteht ein Ausweg darin, dass die betreffenden Anwendungen selbst solche Daten ermitteln und dem Leistungsmanagement explizit zur Verfügung stellen.

Außer der Einhaltung von Dienstgütevereinbarungen dient das Leistungs-management:

- der Führung und Auswertung von Protokollen (Performance Log)
- der Durchführung von Leistungstests
- dem Erkennen von Leistungsengpässen und damit dem Vorbeugen von Problemen
- der Generierung von Berichten und Statistiken zu Netz- und Systembelastungen
- der Gewinnung von Informationen für die Planung nötiger Netz- und System-erweiterungen

2.2.5 Sicherheitsmanagement

Die Dienste, die von einem verteilten System erbracht werden sowie die von diesem System verarbeiteten Informationen, stellen Werte dar, die unterschiedlichen Bedrohungen ausgesetzt sind. Das Spektrum dieser Bedrohungen umfasst u.a.:

- Fehlbedienung
- Fehlfunktion von Ressourcen

- passive Angriffe, dazu zählen das Abhören von Informationen und das Ausspähen von Daten einschließlich Passworten
- aktive Angriffe, wie beispielsweise:
    - Wiedereinspielen von mitgeschnittenen Nachrichtensequenzen
    - Diebstahl von Hard- oder Software
    - Vortäuschen einer falschen Identität
    - Modifikation von Nachrichten bzw. Daten
    - sog. „Denial-of-Service“-Angriffe, die darauf abzielen Dienste und Resourcen durch zielgerichtete Überlastung zu blockieren (Überflutung eines Servers mit Anfragen)
    - Viren

Ziel des Sicherheitsmanagements ist es, die Risiken, die von den vorhandenen Bedrohungen ausgehen, auf ein akzeptables Maß zu beschränken und damit dem Datenverlust und dem Verlust der Vertraulichkeit von Daten vorzubeugen. Aus der Analyse der vorhandenen Bedrohungen müssen die angestrebten Sicherheitsziele abgeleitet werden. Diesbezüglich werden alle grundlegenden Richtlinien und Anforderungen eines konkreten Unternehmens als sog. Sicherheitspolitik dieses Unternehmens zusammengefasst. Eine solche Sicherheitspolitik bildet die Grundlage für das Sicherheitsmanagement und sollte daher explizit formuliert werden. Aus dieser Sicherheitspolitik sollte insbesondere hervorgehen ([Sos00], [Tiv1]):

- welche Daten und Ressourcen wie zu schützen sind und welcher Aufwand dafür betrieben werden darf bzw. muss
- welche Restrisiken in Kauf genommen werden
- wie die Überwachung der Einhaltung der Sicherheitspolitik erfolgt
- wer wofür verantwortlich ist
- was bei Verstößen gegen die Sicherheitsrichtlinien des Unternehmens passiert
- welche Richtlinien zum Zugriffsschutz existieren, und zwar sowohl unter physischen Aspekten (z.B. Computer, die als Server dienen, sind in separaten Räumen unterzubringen) als auch logischen Aspekten (wie die Definition von Sicherheitsstufen und Aufgabenbereichen für Benutzer und Administratoren)
- welche Regeln für den Passwortschutz gelten (z.B. wie oft Passworte gewechselt werden müssen)

Das Sicherheitsmanagement umfasst die Steuerung und Überwachung aller Einrichtungen und Mechanismen, die der Umsetzung der Sicherheitspolitik dienen. Dies betrifft u.a. folgende Maßnahmen:

- Feststellen der Identität, Zugangskontrolle:
  Dazu gehören die Identifikation von Benutzern, was meist durch die Vergabe

einer eindeutigen Benutzerkennung realisiert wird, sowie die Authentisierung der Benutzer gegenüber dem IT-System (bzw. einem Server). Ziel dieser Maßnahmen ist es, unberechtigten Benutzern den Zugang zu den IT-Systemen zu verwehren. Die Authentisierung eines Benutzers dient dem Beweis, dass es sich tatsächlich um den angegebenen Benutzer handelt. Dies geschieht in den meisten Fällen dadurch, dass der Benutzer beim Systemzugang (Login) ein Passwort eingeben muss. Weitere Authentisierungsverfahren beruhen auf dem Besitz einer Chipkarte oder der Prüfung biometrischer Merkmale des Benutzers (wie Fingerabdruck, Regenbogenhaut, Gesichtserkennung). Authentisierungsverfahren, bei denen auch ein Server seine Identität gegenüber dem Benutzer bestätigen muss, sind in der Praxis derzeit eher selten zu finden, werden aber nach [Cyp92] mit der weiteren Verbreitung verteilter Systeme an Bedeutung gewinnen.

Eine verbreitete Forderung an Authentisierungsverfahren in verteilten Systemen besteht darin, dass der Benutzer nach erfolgter Authentisierung sämtliche ihm zustehenden Ressourcen nutzen kann, ohne sich nochmals (bei mehreren von ihm genutzten Servern) anmelden zu müssen. Diese Eigenschaft wird als „Single-Sign-On“ bezeichnet und stellt insbesondere in heterogenen IT-Landschaften eine Herausforderung für das Sicherheitsmanagement dar.

- Benutzerverwaltung:
  Diese umfasst das Anlegen, Pflegen und Löschen von Benutzerkennungen und Benutzergruppen, die Verwaltung von Mitgliedschaften in Benutzergruppen, die Erteilung und Verwaltung von Passworten sowie die Planung von Verzeichnisstrukturen. Die Vergabe von Benutzerkennungen, die auch als Benutzernamen bezeichnet werden, erfolgt gegebenenfalls nach einem unternehmensinternen Schema. Zum einen sind die Unterschiede der beteiligten IT-Systeme hinsichtlich der maximal zulässigen Länge für Benutzernamen zu berücksichtigen. Andererseits ist es oft wünschenswert, anhand des Benutzernamens organisatorische Informationen, wie die Zugehörigkeit zu Benutzerklassen (z.B. entsprechend der Abteilungszugehörigkeit), ableiten zu können. Die Definition und Überwachung diesbezüglicher Richtlinien gehört ebenfalls zu den Aufgaben der Benutzerverwaltung. Die Benutzerverwaltung muss außerdem auf Veränderungen reagieren, beispielsweise wenn ein Mitarbeiter innerhalb des Unternehmens umzieht, seine Chipkarte verliert oder das Unternehmen dauerhaft verlässt.
- Zugriffskontrolle (Access Control) für die Nutzung von Ressourcen (Autorisation):
  Hier geht es darum, sicherzustellen, dass (insbesondere sensible) Daten und Ressourcen nur von berechtigten Personen genutzt werden können. In vielen Unternehmen ist es nötig, den Mitarbeitern einen unkomplizierten, benutzerfreundlichen Zugang zu einer Fülle von Informationen zur Verfügung

zu stellen, um ein effizientes Arbeiten zu ermöglichen. Dabei kann auf Informationen in Dateien, Datenbanken oder anderen Informationssystemen zugegriffen werden. Der Wunsch nach Einfachheit beim Zugang zu Informationen kann allerdings sehr leicht mit einer Bloßstellung der Sicherheit einhergehen. Eine Möglichkeit, um derartigen Begleiterscheinungen vorzubeugen, ist die Nutzung von speziellen Systemen zur Zugriffskontrolle (Access Control Systems). Mit ihnen können die Zugriffsberechtigungen der Benutzer auf IT-Ressourcen (Programme, Daten, Dienste) spezifiziert werden. Das Ziel besteht darin, den Benutzern nur Zugriff auf Daten zu gewähren, die sie bei ihrer Arbeit benötigen und darüber hinaus die Zugriffsberechtigungen auf das tatsächlich erforderliche Maß zu beschränken. Derartige Systeme zum Zugriffsschutz gibt es in verschiedenen Ausprägungen, denen unterschiedliche Sicherheits-Modelle zugrunde liegen können. Verbreitet und praktikabel sind sog. „Discretionary Access Control Systems“. In solchen Systemen bestimmt der Besitzer einer Ressource, wer auf diese Ressource zugreifen darf und welche Zugriffsart erlaubt wird. Eine Erweiterung dieser Systeme stellen „Mandatory Access Control Systems“ (MAC-Systems) dar. Bei ihnen werden zusätzlich Ebenen der Vertraulichkeit von Daten eingeführt. Sämtliche Daten werden entsprechend ihrer (Un)bedenklichkeit deklariert und in bestimmte Zugriffsklassen eingeordnet. Jeder Nutzer, dessen Privilegien gleich oder größer der Zugriffsklasse der Daten sind, darf diese Daten lesen. Das Verteidigungsministerium der USA benutzt beispielsweise diesen Typ von Zugriffsschutzsystemen [Cyp92]. Die Klassifikation der Daten stellt allerdings eine Hürde bei der Umsetzung eines solchen Systems dar. Viele Unternehmen verzichten daher auf MAC-basierte Systeme. Weitere Ausführungen zu Sicherheitsmodellen sind u.a. in [Sos00] und [Bub96] zu finden.

- automatische Überwachung der IT Nutzung um Sicherheitsverstöße (möglichst bereits beim Versuch) zu erkennen
- Führen und Auswerten von Protokollen (security logs) für sicherheitsrelevante Ereignisse
- Erstellung von Berichten über Berechtigungen, Historien und Schutzverletzungen
- Einsatz und Betrieb von Firewalls:
  Ein Firewall-System dient als einziger Verbindungspunkt zwischen zwei Netzen. Die Aufgabe einer Firewall besteht darin, zwischen diesen Netzen nur vorgesehenen, d.h. erlaubten, Datenverkehr zuzulassen. Nähere Informationen zu diesem komplexen Thema sind z.B. in [Che96] zu finden.
- Einsatz von Verschlüsselungstechniken:
  Dies betrifft u.a. die Generierung, Verteilung und Speicherung von Verschlüsselungscodes sowie die Überwachung ihrer Gültigkeitsdauer.

Eine grundsätzliche Schwierigkeit des Sicherheitsmanagements wird durch die Heterogenität der vorhandenen Systeme verursacht. Es muss die „individuellen“, das heißt inkompatiblen, Sicherheitssysteme der eingesetzten Anwendungsprogramme und Betriebssysteme steuern. Meist müssen Benutzer auf mehreren Servern und Systemen, wie beispielsweise Mailservern und Datenbanksystemen, eingerichtet und gepflegt werden. In der Praxis gewinnen deshalb Verzeichnisdienste (Directory Services) an Bedeutung. Sie können Informationen (z.B. Benutzerinformationen wie Benutzername, Adresse usw.) in einer baumartigen, hierarchisch aufgebauten Struktur systemübergreifend mehreren Anwendungen bzw. Servern zur Verfügung stellen. Prinzipiell kann die komplette Organisationsstruktur eines Unternehmens in ein derartiges Verzeichnis abgebildet werden. Leider sind auch die bisher existierenden Verzeichnisdienste nicht miteinander kompatibel und teilweise nur auf wenige Betriebssystemplattformen beschränkt [Kup98]. Als gemeinsamer Nenner für den Zugriff auf Verzeichnisdienste hat sich das standardisierte Zugangsprotokoll LDAP (Lightweight Directory Access Protocol) durchgesetzt. Allerdings müssen die eingesetzten Anwendungen und Server auch von einem Verzeichnisdienst Gebrauch machen können, indem sie beispielsweise über LDAP auf einen Verzeichnisdienst zugreifen, dies ist derzeit meist nicht der Fall.

2.3 Konzept und Arbeitsweise von Managementsystememen

In Abbildung 2 ist das Manager-Agent-Paradigma dargestellt, das den meisten Managementsystemen als Funktionsprinzip zugrunde liegt. Es findet sich sowohl beim SNMP-Management als auch in den ISO-Standards zum OSI- Management.

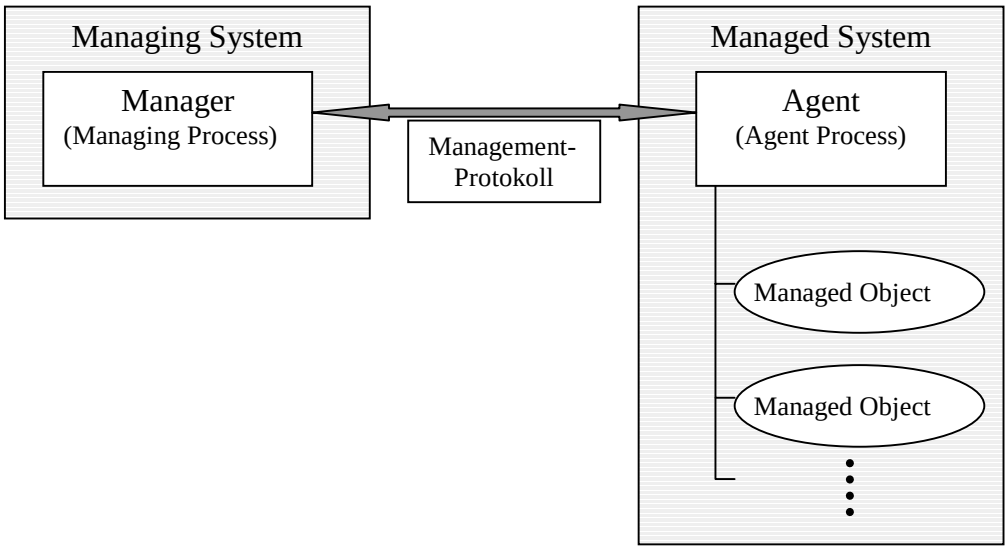

Abbildung 2: grafische Darstellung der Zugriffe im Manager-Agent-Paradigma

Netz- oder Systemkomponenten werden abstrakt in Form von Managed Objects (deutsch etwa: verwalteten bzw. verwaltbaren Objekten) verwaltet, unabhängig davon ob es sich bei den Komponenten um physische Ressourcen wie Router, Workstations oder logische Ressourcen wie z.B. Verbindungsinformationen handelt. Ein Managed Object, siehe Abb. 3, besitzt ein oder mehrere Attribute, die den Eigenschaften der realen Ressourcen entsprechen, die verwaltet werden sollen. Auf diese Attribute können Operationen angewandt werden, um den Status der Attribute abzufragen oder zu setzen. Das Managed Object kann seinerseits Meldungen erzeugen.

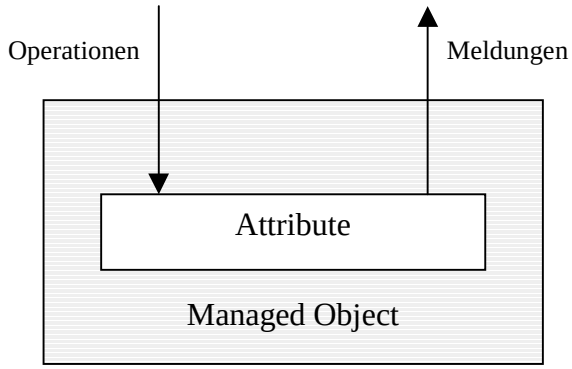

Abbildung 3: grafische Darstellung der Zugriffe auf ein Managed Object [Lex94]

Gegenüber einem Managementsystem werden Managed Objects durch einen Agenten vertreten. Der Agent residiert in dem System, das verwaltet wird (englisch: Managed System). Das Gegenstück zur Rolle des Agenten ist der Manager. Beide kommunizieren miteinander über das Managementprotokoll. Sowohl

beim Agenten als auch beim Manager handelt es sich um Softwareinstanzen, weshalb sie in der Literatur auch als Agent-Prozess (englisch: Agent Process) bzw. Manager-Prozess (englisch: Managing Process) bezeichnet werden. Eine wesentliche Aufgabe des Managers besteht darin, im Auftrag des Managementsystems mit mehreren Agenten zu kommunizieren, um Management-Daten für die Auswertung im Managementsystem zu sammeln und Operationen an die Agenten weiterzuleiten (siehe Abb. 4). In den meisten Fällen werden die Managementinformationen in dem gleichen Netz transportiert, wie die eigentlichen Anwendungsdaten. Diese Vorgehensweise heißt „Inband-Management“, während die Verwendung eines separaten Netzwerkes für die Übertragung der Managementinformationen als „Outband-Management“ bezeichnet wird [Jan93].

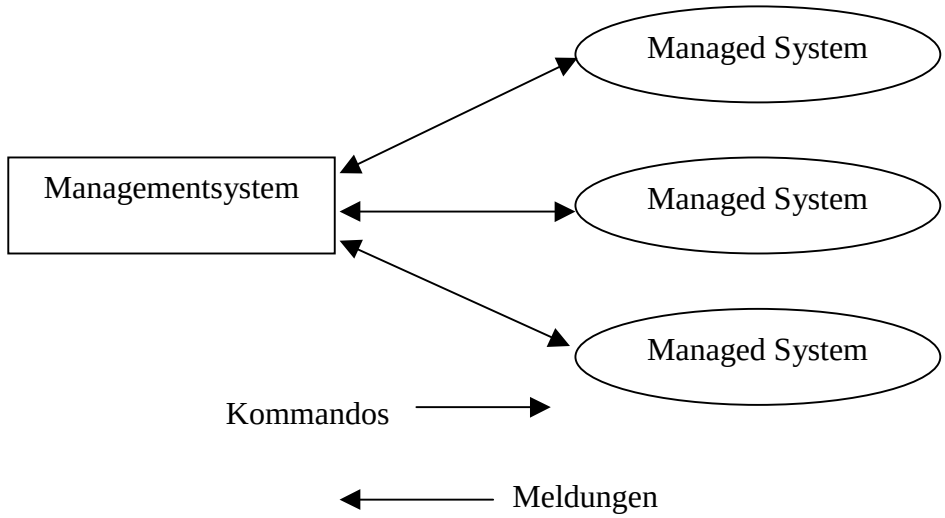

Abbildung 4: grafische Darstellung des Austauschs von Managementinformationen (nach [Jan93])

Die generelle Arbeitsweise von Managementsystemen lässt sich durch die folgenden 6 Schritte beschreiben [Cyp92]:

1. Sammeln von Statusinformationen der Netz- und Systemkomponenten
2. Umwandlung dieser Informationen in ein vorgegebenes Format
3. Transport der Informationen

4.-6. Speichern, Analysieren und Reagieren

Für die Abarbeitung dieser Schritte müssen beim Entwurf eines Management-systems u.a. Probleme allgemeiner Natur gelöst werden. So verlangt Schritt 1 neben der Managementfähigkeit der Komponenten – d.h. der Existenz eines Agenten in diesen Komponenten – auch die Definition von Ressourceninformationen wie z.B. Objektattributen. Die Schritte 2 und 3 erfordern sowohl Festlegungen zur

Syntax, nach denen die Ressourceninformationen kodiert und dekodiert werden, als auch zum Management-Protokoll und zur Kommunikationsarchitektur:

- Wie erfolgt die Kommunikation zwischen Manager und Agent?
- Gibt es mehrere Manager?
- Wie erfolgt gegebenenfalls die Kommunikation zwischen Managern?
- Welche Strukturen/Hierarchien lassen sich aufbauen?
- Wie sieht die Zuordnung der Rollen „Agent“ und „Manager“ aus? Handelt es sich um eine starre Zuordnung oder ist ein Wechsel je nach Kontext möglich, d.h. kann z.B. ein Manager auch Agent sein?

Um nicht „jedes Mal das Rad neu erfinden zu müssen“, wurde und wird von verschiedenen Gremien und Organisationen – stellvertretend seien hier ISO und IAB (Internet Architecture Board) genannt – versucht, Empfehlungen und Standards zu entwickeln, die Probleme des Aufbaus und der Funktionsweise von Managementsystemen direkt oder indirekt adressieren. Die Erarbeitung derartiger Festlegungen ist meist ein recht langwieriger Prozess, der durch teilweise kontrovers geführte Diskussionen versucht, zu einem Konsens zu gelangen. Für eine Reihe von Problemen, die hauptsächlich die Schritte 1 bis 3 betreffen, wurden Lösungen erarbeitet und entsprechende Standards oder Empfehlungen verabschiedet. Allerdings konnte sich bisher nicht jeder erarbeitete Standard auch tatsächlich am Markt durchsetzen. Die derzeit verfügbaren Managementsysteme sind zu einem wesentlichen Teil durch proprietäre Lösungen gekennzeichnet. Dafür gibt es mehrere Ursachen, wie beispielsweise:

- fehlende oder unzureichende Standards
- die Innovation der Hersteller bei der Verbesserung ihrer Produkte
- der Einfluss neuer Technologien
- natürliche Herstellerinteressen: Das eigene Produkt soll am Markt etabliert werden, es soll daher individuelle Vorzüge aufweisen und darf nicht ohne weiteres gegen ein Konkurrenzprodukt austauschbar sein.

Im praktischen Einsatz von Software zum NSM besteht daher die Notwendigkeit der Integrationsfähigkeit unterschiedlicher Produkte. Mögliche Ansatzpunkte zur Integration sind insbesondere:

- ein gemeinsames Management-Protokoll
- zumindest ein gemeinsamer Teilbereich des verwendeten Datenmodells
- die Existenz von offengelegten APIs (Anwendungsprogrammierschnittstellen) der Managementsysteme

Eine Technik, die sich auch in der Praxis durchsetzen konnte, ist das Internet-Management auf der Basis von SNMP, das im Folgenden erläutert werden soll.

2.4 SNMP

2.4.1 Zur Entstehung

Das SNMP-Management wurde entwickelt, um die Verwaltung von TCP/IP-basierten Netzwerken zu ermöglichen. Hintergrund dieser Entwicklung war die Notwendigkeit, das Management von Komponenten im Internet effektiver zu gestalten. Es sollte ein herstellerneutrales Verfahren geschaffen und standardisiert werden. Dazu kamen Anfang 1988 beim IAB drei verschiedene Lösungsansätze in die engere Wahl und wurden näher untersucht [RFC1052]:

- High-Level Entity Management System (HEMS)
- Simple Gateway Monitoring Protocol (SGMP)
- Common Management Information Services/Common Management Information Protocol (CMIS/CMIP); hierbei handelt es sich um eine Initiative der ISO

Der HEMS-Vorschlag wurde im Interesse einer Harmonisierung der weiteren Entwicklung zurückgezogen. Der CMIS/CMIP-Ansatz deckte konzeptionell den weitesten Bereich ab. Die zugehörigen Spezifikationen waren allerdings noch nicht präzise definiert, eine Beispielumsetzung des Konzeptes fehlte. Da ein dringender Bedarf an Lösungen für den praktischen Einsatz bestand, wurde vom IAB eine zweigleisige Strategie für die weitere Entwicklung gewählt. Um kurzfristig Netzwerkmanagementlösungen in die Praxis überführen zu können, wurde vorgesehen, die bisherige SGMP-Entwicklung unter der neuen Bezeichnung SNMP weiterzuverfolgen. Langfristig sollte ein einheitliches Verfahren für das Management von Netzwerken entwickelt werden, das auf CMIS/CMIP basiert. Der dabei angestrebte Standard sollte die Übergangslösung SNMP ergänzen oder gar ersetzen. Es wurde angenommen, dass sich künftig die OSI-Protokolle der ISO etablieren und damit auch die TCP/IP-Protokollfamilie verdrängen würden. Während sich SNMP nach kurzer Zeit weit verbreitete und greifbare Ergebnisse lieferte, erfüllten sich die in die OSI-Entwicklung gesetzten Erwartungen nicht. Die folgenden Ausführungen beziehen sich auf die erste Version des SNMP-Managements SNMPv1.

2.4.2 Modell und Konzept

SNMP bezeichnet eigentlich nur das Management-Protokoll, allerdings werden mit dieser Bezeichnung auch das zugehörige Konzept und Verfahren assoziiert. Außer dem Management-Protokoll gehören noch Managementinformationen, mindestens eine Netzwerkmanagementstation (NMS) und Netzwerk-Elemente

zur grundlegenden Architektur. Netzwerk-Elemente, teils auch als Netzwerk-Knoten (Network Nodes) häufiger jedoch als Managed Nodes (verwaltete Knoten) bezeichnet, werden die Geräte genannt, die verwaltet werden sollen. Sie besitzen je einen Agenten. Die Netzwerkmanagementstationen führen Management-Anwendungen aus, die für die Überwachung und Steuerung der Netzwerk-Elemente sowie die Auswertung und Präsentation der gewonnenen Daten zuständig sind. SNMP dient zur Übertragung der Managementinformationen zwischen NMS und Agenten.
Vorrangige Ziele beim Entwurf von SNMP waren zum einen die Einfachheit („Simple“) und zum anderen die Erweiterbarkeit dieses Managementkonzeptes. Ersteres wurde als sog. „Fundamental Axiom“ formuliert (nach [Ros94, S.62]):

> *„Die durch das Netzwerkmanagement zusätzlich entstehenden Belastungen der Managed Nodes müssen minimal sein, also den kleinsten gemeinsamen Nenner widerspiegeln.“*

Die Einfachheit betrifft also insbesondere die Agenten. Sie müssen von den Geräteherstellern zusätzlich in die betreffenden Produkte integriert werden. Dies erfordert natürlich zusätzliche Ressourcen (beispielsweise Zähler), die nur für die Managementfähigkeit der Geräte erforderlich sind und zu höheren Kosten führen. Durch das „Fundamental Axiom“ wird die Komplexität des Managements schwerpunktmäßig auf die Managementstationen verlagert. Das Konzept des SNMP-Managements basiert auf 3 Säulen :

1. Festlegungen zu Struktur und Identifizierung von Managementinformation [RFC1155]
2. Einer Anfangsmenge von Managed Objects für TCP/IP-basierte Netzwerke [RFC1213]
3. dem Management-Protokoll SNMP [RFC1157]

2.4.2.1 Managementinformation

Die Erweiterbarkeit von SNMP kommt hauptsächlich in den Festlegungen zu Managementinformationen zum Ausdruck. Managementinformationen werden in Form von Managed Objects repräsentiert. Während die Regeln, nach denen Managed Objects definiert werden, strikt festgelegt sind, kann die Anfangsmenge der Managed Objects ergänzt werden. Die Objektklassen und Instanzen aller Managed Objects sind in einer virtuellen Datenbasis, der so genannten „Management Information Base“ (MIB), enthalten. Die MIB ist sowohl in der Netzwerkmanagementstation als auch in den Managed Nodes vorhanden, siehe Abbildung 5.

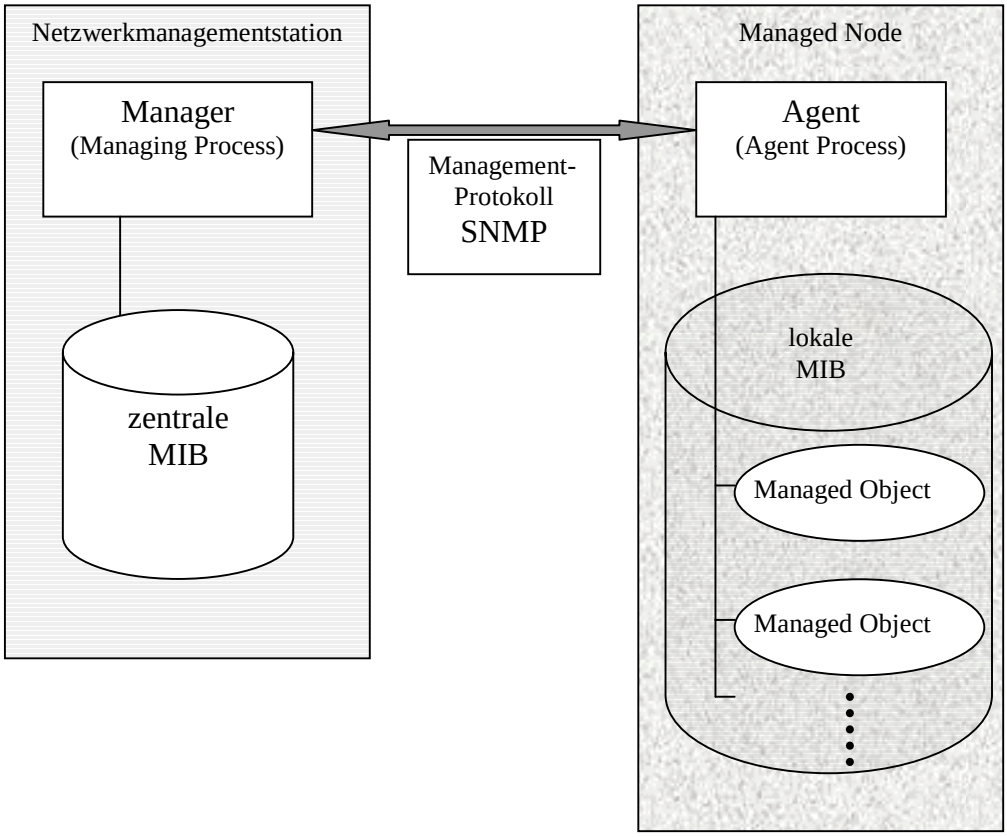

Abbildung 5: grundlegendes Modell des SNMP-Managements

Die MIB auf Seiten eines Agenten enthält nur die Managed Objects, die lokal von dem betreffenden Managed Node unterstützt werden. Die zentrale MIB in der Netzwerkmanagementstation umfasst hingegen alle Managed Objects die in den (von dieser NMS aus verwalteten) Managed Nodes vorkommen.
Thematisch zusammengehörende Managed Objects werden zu einem MIB-Modul zusammengefasst und als solches standardisiert. Innerhalb der MIB-Module sind die Managed Objects in Gruppen angeordnet. Von den Managed Nodes müssen die unterstützten Managed Objects gruppenweise implementiert werden, das heißt entweder alle Objekte der betreffenden Gruppe werden unterstützt oder keines von ihnen. Die MIB in einem Managed Node kann Gruppen aus verschiedenen MIB-Modulen enthalten. Eine Anfangsmenge von Managed Objects für Managed Nodes in TCP/IP-basierten Netzen wurde als „Standard-MIB“ definiert und in einem MIB-Modul zusammengefasst. Eine neue Version der Standard-MIB kann ihre Vorgängerversion durch Hinzufügen neuer Managed Objects ergänzen. Diese Standard-MIB liegt derzeit in der zweiten Version vor, definiert in [RFC1213], und wird deshalb als „MIB-II“ bezeichnet. Die Gruppen der MIB-II sollten von allen Agenten unterstützt werden, es sei denn eine Gruppe bezieht

sich auf ein Protokoll (wie z.B. EGP), welches in dem betreffenden Managed Node nicht unterstützt wird.
Die Managed Objects werden mit Hilfe der formalen Datenbeschreibungsprache „Abstract Syntax Notation One“ (ASN.1) definiert. ASN.1 wurde von der ISO entwickelt (ISO8824), um beim Datenaustausch zwischen zwei End-Systemen eine gemeinsame abstrakte, systemunabhängige Sprache zur Definition der zu übertragenden Daten zu haben. Grundsätzlich ist ASN.1 eine sehr mächtige Sprache, mit der einfache Objekte zu komplexeren Objekten zusammengesetzt werden können. ASN.1 bietet für die Definition von Datenstrukturen ähnliche Möglichkeiten wie Programmiersprachen [Sta93]. Objekte können einfache oder zusammengesetzte Datentypen, Variablen oder Makros enthalten. Die „Basic Encoding Rules“ (BER) von ASN.1, definiert in der ISO-Norm 8825, legen fest, wie ASN.1-Variablen bei der Übertragung über das Netz serialisiert, d.h. als Bytestrings codiert bzw. auf dem Zielsystem wieder rekonstruiert, werden. Das SNMP-Management verwendet zur Beschreibung der Managementinformationen allerdings nur eine Untermenge von ASN.1 [Jan93]. Die Managed Objects stellen hier keine vollwertigen Objekte im Sinne der objektorientierten Programmierung dar. Es ist kein Vererbungskonzept definiert. Ein Managed Object hat nicht beliebig viele Attribute, es repräsentiert im Wesentlichen eine Variable, beschreibt also (genau) eine bestimmte Eigenschaft, ein Attribut eines Netzwerk-Elements.
Zur Definition der Managed Objects werden hauptsächlich die einfachen ASN.1-Datentypen „INTEGER“; „OCTET STRING“ und „OBJECT IDENTIFIER“ sowie die zusammengesetzten Typen „SEQUENCE“ und „SEQUENCE OF“ verwendet, deren Bedeutung aus Tabelle 1 hervorgeht (nach [Jan93]).

ASN.1-Datentyp	Bedeutung
INTEGER	Ganzzahl
OCTET STRING	eine Folge von 0 oder mehr Bytes, jedes Byte kann einen Wert von 0 bis 255 annehmen
OBJECT IDENTIFIER	dient der Benennung von Objekten
SEQUENCE	Eine Liste von 0 oder mehr Elementen, die Listenelemente haben einen einfachen Datentyp
SEQUENCE OF	Eine Liste von Elementen gleichen Typs

Tabelle 1: einige ASN.1-Datentypen, die das SNMP-Management verwendet

Mit Hilfe von „SEQUENCE“ und „SEQUENCE OF“ lassen sich Tabellen modellieren. Außerdem können aus den verwendeten ASN.1-Datentypen neue Typen (Subtypes) abgeleitet werden, die gegebenenfalls nur einen eingeschränkten Wertevorrat besitzen.

Die Objektklassen werden beim SNMP-Management als Objekttypen bezeichnet, während die Objektinstanzen auch Variablen genannt werden. Jeder Objekttyp ist definiert durch:

- seinen Namen (bestehend aus einem Objektidentifikator und einer zugehörigen textlichen Beschreibung, dem Objektdeskriptor)
- seine Syntax (die Datenstruktur der Variablen)
- die zugehörige Kodierung (entsprechend ASN.1 BER)
- eine Beschreibung der Semantik
- einer Angabe des möglichen Zugriffs (entweder: Lesen, Schreiben, Lesen und Schreiben, kein Zugriff)
- einer Statusinformation, ob die Implementierung dieses Objekttyps verbindlich, optional oder obsolet ist

Der Name wird duch einen so genannten „ASN.1-Object-Identifier“ festgelegt. Diese Object-Identifier (OID) werden auf administrativem Wege zentral vergeben. Die ISO und die ITU (International Telecommunication Union) verwalten gemeinsam einen Registrierungsbaum, der alle vergebenen ASN.1-Object-Identifier enthält. Außer dem Wurzelknoten besitzt jeder Knoten dieses Baumes eine Markierung in Form einer nichtnegativen Integer-Größe sowie eine textuelle Beschreibung. Der OID eines Baumknotens entsteht durch das Aneinanderreihen der Markierungen aller Knoten, die auf dem Pfad vom Wurzelknoten bis zu dem betreffenden Baumknoten liegen, dabei werden die Markierungen durch einzelne Punkte getrennt. Der Wurzeknoten hat die drei Nachfolgeknoten „itu-t“, „iso“ und, „joint-iso-itu“, siehe Abbildung 6.

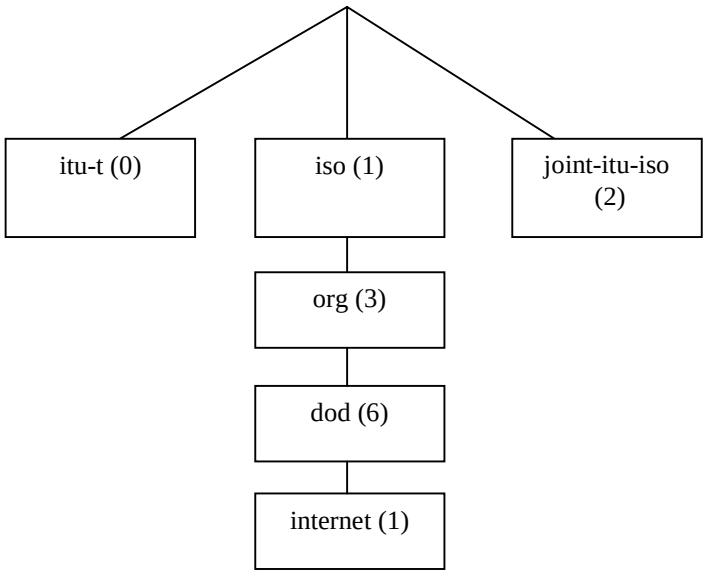

Abbildung 6: Registrierungsbaum für ASN.1-Objekte (nach [Heg99])

Der Registrierungsbaum dient nicht nur zur Benennung von Managed Objects, so werden beispielsweise internationale Standardisierungsdokumente ebenfalls über OIDs im Registrierungsbaum identifiziert. Die ISO stellt in dem von ihr administrierten Teil des Baumes („iso“) auch anderen Organisationen einen Zweig zur Verfügung, unter dem diese OIDs verwalten können. Das Verteidigungsministerium der USA (DoD) hat in diesem Zweig den Baumknoten mit der Nummer 6 erhalten. Ausgangspunkt für die Verwaltung von OIDs durch das IAB ist der Koten mit der Bezeichnung „1.3.6.1“ bzw. „iso.org.dod.internet“ [RFC1155]. Für das SNMP-Management sind die drei Nachfolgeknoten „mgmt“, „experimental“ und „private“ des Knoten „internet“ von besonderem Interesse, siehe Abbildung 7.

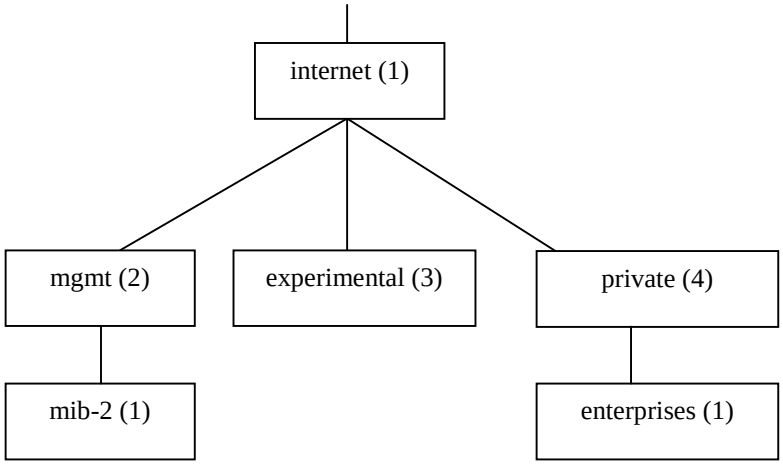

Abbildung 7: Ausschnitt des von der IAB verwalteten Teilbaumes

Der Knoten „mgmt“ dient als Anfangsknoten eines Teilbaumes zur Identifizierung aller Managed Objects, die in Standardisierungsdokumenten durch das IAB genehmigt wurden. Direkter und derzeit einziger Nachfolgeknoten von „mgmt“ ist der Knoten „mib-2“. Unter „mib-2“ sind zunächst die insgesamt elf Gruppen des MIB-Moduls MIB-II angeordnet. Unterhalb dieser Gruppen befinden sich die Managed Objects der MIB-II. Parallel zu den Gruppen der MIB-II werden Knoten für weitere MIB-Module, die die Anfangsmenge der Managed Objects der MIB-II ergänzen, eingeordnet.
Der Bereich unter „experimental“ beherbergt Managed Objects, die im Rahmen von Internet-Experimenten für Test- und Entwicklungszwecke registriert werden. Der Teilbaum unterhalb des „private“-Knoten dient zur Registrierung von MIBs, die außerhalb des IAB festgelegt werden und herstellerspezifische Managed Objects enthalten. Daher befindet sich unter „private“ nur der Knoten „enterprises“, unter dem die Knoten zur Identifizierung von Herstellerfirmen angeordnet sind. Die Unternehmen können die Teilbäume unterhalb ihres Knoten nach eigenem

Ermessen gestalten, obwohl in [RFC1155] eine offizielle Registrierung der herstellerspezifischen Objekte ausdrücklich empfohlen wird.
Wie auf die Instanzen von Managed Objects in der MIB zugegriffen wird, ist von dem verwendeten Managementprotokoll festzulegen.

2.4.2.2 Managementprotokoll

Laut [RFC1155] muss das Managementprotokoll einen Mechanismus für den Zugriff auf Instanzen einfacher Objekttypen definieren. Ein Zugriffsmechanismus auf Instanzen zusammengesetzter Objekttypen ist optional. Das Managementprotokoll SNMP schließt den Zugriff auf zusammengesetzte Objekttypen aus, es bietet nur den Zugriff auf Instanzen einfacher Objekttypen [RFC1157]. Für Objekttypen von denen es im Managed Node nur eine Instanz gibt, wird diese durch den OID des Objekttyps und eine angefügte „0“ identifiziert. Repräsentiert der Objekttyp hingegen eine Tabelle, so wird der OID um einen Suffix erweitert, der für die Identifikation jeder einzelnen Tabellenzelle sorgt.
In der MIB eines Managed Node sind die Instanzen der Managed Objects entweder vorhanden oder nicht. SNMP definiert keine Operationen um Managed Objects zu erzeugen oder zu vernichten. Bei SNMP handelt es sich um ein „Request-Response-Protokoll“, das über einen verbindungslosen Transportdienst Nachrichten zwischen Agent und Manager befördert. Der Manager sendet dem Agenten Anforderungen (Requests), um Variablenwerte in der MIB des Netzwerk-Elementes zu lesen oder zu schreiben. Der Agent führt die gewünschte Aktion aus und sendet an den Manager eine Antwort (Response). Das Managementprotokoll SNMP definiert fünf Operationen, siehe Tabelle 2.

Operation	Senderichtung	Bedeutung
Get-Request	Manager an Agent	Abfrage von angegebenen Variablen
Get-Next-Request	Manager an Agent	Abfrage von (lexikographisch) nächsten Variablen in der MIB, damit kann die MIB eines Managed Node sukzessiv gelesen werden
Set-Request	Manager an Agent	setzt den Wert von Variablen
Get-Response	Agent an Manager	als Anwort auf Get, Get-Next und Set, liefert die angeforderten Daten bzw. die neu gesetzten Werte
Trap	Agent an Manager	dient zur Übermittlung von spontanen Meldungen, um aufgetretene Ereignisse anzuzeigen

Tabelle 2: Operationen des Managementprotokolls SNMP

Jede dieser Operationen wird in einer entsprechenden Protokolldateneinheit, einer so genannten „Protocol Data Unit“ (PDU), eingebettet. Die Operationen „Get-Request“, „Get-Next-Request“, „Set-Request“ und „Get-Response“ beziehen sich jeweils auf eine Liste von Name-Wert-Paaren der Objektinstanzen. Die zugehörigen Lese- oder Schreiboperationen sind dabei atomar, d.h. falls eine Variable der Liste nicht gelesen oder verändert werden kann, so wird die Lese- oder Schreiboperation für alle Variablen in der Liste verweigert. Für die Trap-Operation wird keine Rückantwort bzw. Empfangsbestätigung gesendet. Als Reaktion auf das signalisierte Ereignis werden durch die Management-Anwendung auf Seiten der NMS gegebenenfalls weitere SNMP-Operationen ausgelöst, diese Vorgehensweise wird auch als „Trap-Directed Polling“ bezeichnet. Außer einer PDU sind noch eine Versionsangabe und ein Community-Name Bestandteil einer SNMP-Nachricht [RFC1157]. Jede Nachricht wird als ein Datenpaket versendet. SNMP nutzt dazu das Transportprotokoll UDP, siehe Abbildung 8.

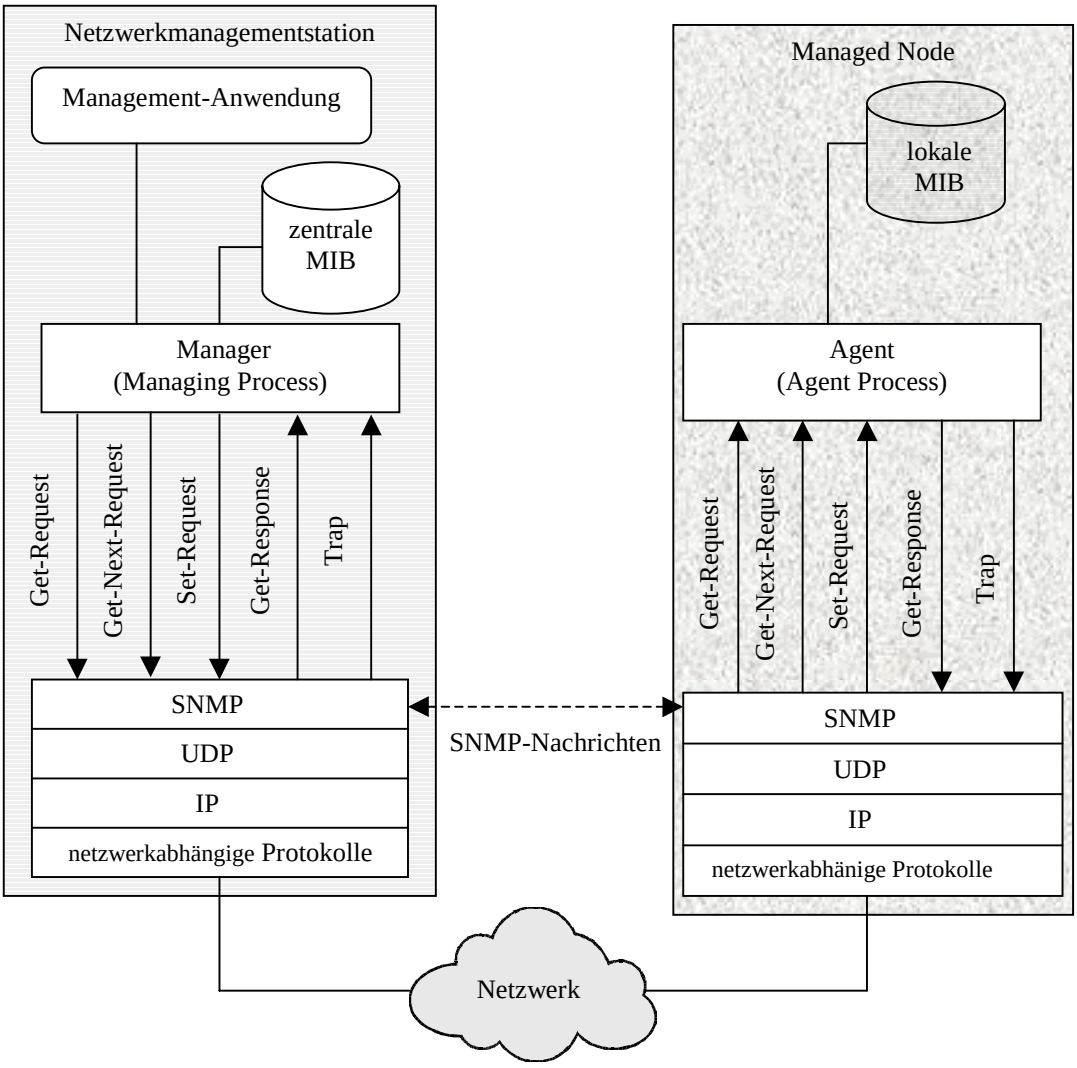

Abbildung 8: schematische Darstellung des Protokollkontexts von SNMP (nach [Sta93])

Geräte, die nicht direkt per SNMP ansteuerbar sind, können über einen Proxy-Agenten verwaltet werden. Der Proxy-Agent arbeitet als Stellvertreter für diese Geräte. Er kommuniziert mit der NMS wie jeder andere Agent mittels SNMP und konvertiert die Informationen in ein gerätespezifisches Protokoll.

2.4.2.3 Sicherheitskonzept

Das Sicherheitskonzept von SNMPv1 basiert auf dem so genannten Community-Konzept. Eine Community ist eine Gruppe, der ein Agent und ein oder mehrere Manager angehören. Die Community wird lokal beim Agenten definiert. Die Community besitzt einen eindeutigen Namen. Dieser Community-Name ist Bestandteil jeder SNMP-Nachricht. Der Agent akzeptiert nur SNMP-Operationen, wenn der Manager zu einer Community gehört, die dem Agenten bekannt ist. Der Community-Name dient also als Passwort. Da in [RFC1157] kein verbindliches Verschlüsselungsverfahren für den Community-Namen vorgeschrieben wurde, wird er in der Praxis meist unverschlüsselt übertragen [Jan93]. Zu jeder Community wird ein Comunity-Profile festgelegt. Durch dieses Community-Profile lässt sich spezifizieren, auf welche Managed Objects in der lokalen MIB auf Seiten des Agenten zugegriffen werden kann. Es besteht aus einer beliebigen Teilmenge von Managed Objects dieser MIB, der so genannten MIB-View, und einer Zugriffsberechtigung (Access Mode). Diese Zugriffsberechtigung kann die Werte „Nur-Lesen“ oder „Lesen und Schreiben“ annehmen und gilt für die gesamte MIB-View. Ob auf ein Managed Object zugegriffen werden darf, ist abhängig vom eingestellten Community-Profile und der Zugriffsangabe bei der Definition des Objkekttyps dieses Managed Objects. Pro Agent können mehrere Communities angelegt werden, wobei durchaus gleiche Manager Mitglieder in unterschiedlichen Communities sein können.

2.4.3 Probleme und weitere Entwicklung

Eine gravierende Schwäche der ersten Version des SNMP-Managements war das unzureichende Sicherheitskonzept. Es bot insbesondere keinen ausreichenden Schutz vor:

- dem Ausspähen der übertragenen Informationen
- dem Wiedereinspielen von mitgeschnittenen Nachrichten
- der Modifikation von Nachrichten
- dem Vortäuschen einer falschen Identität

Dies führte dazu, dass SNMP vorwiegend für das Überwachen (Monitoring) und weniger für das Steuern der Netz- und Systemkomponenten eingesetzt wurde. Einige

Hersteller implementierten in ihren Geräten keine Set-Operation [Ros94]. Neben den sicherheitsbezogenen Problemen von SNMPv1 gab es weitere Mängel [Sta93]:

- Große Datenmengen (beispielsweise Tabelleninhalte) lassen sich nicht effizient ermitteln, der Manager sendet ein Datenpaket pro Anfrage und erhält als Antwort nur ein Datenpaket.
- Es wird keine Kommunikation zwischen Managern unterstützt, es lassen sich keine Hierarchien aufbauen.
- Es ist nicht geeignet für die Verwaltung großer Netze mit vielen Managed Nodes, weil durch den Abfragebetrieb zu viel Datenverkehr im Netzwerk erzeugt wird.
- Da Traps nicht bestätigt werden, können wichtige Ereignismeldungen verloren gehen (durch den verbindungslosen Transportdienst ist der Verlust von Datenpaketen möglich).
- Die MIB-Modellierung ist eingeschränkt, da komplexe Datentypen mit zugehörigen Methoden nicht unterstützt werden.

Bei der Weiterentwicklung von SNMPv1 wurde zunächst am Sicherheitskonzept gearbeitet. Als die entsprechenden Vorschläge beim IAB vorgelegt wurden, gab es bereits Bemühungen, um die anderen Probleme von SNMPv1 zu adressieren. Infolgedessen sollten die Verbesserungen zu einer neuen Version SNMPv2 zusammengefasst werden. Insbesondere die Entwicklungsvorschläge zum Sicherheitskonzept stießen bei der Industrie auf Ablehnung und wurden als zu komplex bewertet [Kra98]. Daraufhin wurde am Community-Konzept festgehalten. Zu den akzeptierten Verbesserungen zählen unter anderem:

- die Get-Bulk-Operation für ein effizienteres Lesen von Tabellen
- die Inform-Operation für Ereignismeldungen, die bestätigt werden
- der Einführung einer Kommunikation zwischen Managern
- eine verbesserte Fehlerbehandlung: Neue Fehlercodes wurden eingeführt. Kann beispielsweise bei der Get-Operation für eine Variable kein Wert ermittelt werden, zeigt dies ein Fehlercode an, die Werte der restlichen Variablen werden aber geliefert.

Da SNMPv1 und SNMPv2 nicht interoperabel sind, wurde eine Koexistenz vorgesehen. Diese beruht entweder auf der Basis des Proxy-Konzeptes oder auf einem Manager, der beide Protokollversionen unterstützt [Ros94]. Weitere Sicherheitskonzepte wurden vorgeschlagen, konnten sich aber nicht auf breiter Basis durchsetzen. Mittlerweile liegt ein Vorschlag für die dritte Version des SNMP-Managements vor [RFC2570]. SNMPv3 baut so weit wie möglich auf vorhandenen Spezifikationen und Verbesserungsvorschlägen auf. Dabei definiert SNMPv3 eine modulare Gesamtarchitektur, die sich flexibel erweitern lässt und die Nutzung unterschiedlicher Versionen des Managementprotokolls gestattet [Kra98]. Das Sicherheitskonzept ermöglicht die Einbindung von verschiedenen

Sicherheitsmodellen und mehreren Modellen für den Zugriffsschutz. Die erste vorgeschlagene Ausprägung eines solchen Sicherheitsmodells [RFC2574] nutzt Verschlüsselungs- und Prüfsummenverfahren, um den eingangs genannten Sicherheitsproblemen wirksam zu begegnen. Damit ist eine sichere Set-Operation realisierbar.

SNMP ist ein etablierter Standard, aber für die Erfüllung aller Kriterien eines integrierten NSM (vgl. Abschnitte 2.1 bis 2.3) ist SNMP weder gedacht noch ausreichend! Daher gibt es fortwährend Bestrebungen, weitere Standards zu entwickeln, um die breitgefächerten Anforderungen eines integrierten NSM besser zu befriedigen. Eine in diesem Zusammenhang vielversprechende Technologie bietet der CORBA-Standard. Obwohl nicht speziell auf das NSM ausgerichtet, ist sie für das NSM aus mehreren Gründen interessant:

- sie ermöglicht die Integration von Softwarekomponenten bis hin zur Anwendungsebene, dies gilt insbesondere plattformübergreifend, in heterogenen IT-Landschaften
- sie ist dabei nicht auf eine Programmiersprache beschränkt
- sie ist objektorientiert, es existieren aber Möglichkeiten bestehende und gegebenenfalls nicht objektorientierte Software (Legacy-Anwendungen) in ein CORBA-System einzubinden.
- es handelt sich um einen offenen, herstellerunabhängigen Standard

Die genannten Eigenschaften werden u.a. auch von Managementwerkzeugen gewünscht, in der Praxis derzeit aber häufig vermisst.

3 CORBA

Die Spezifikation der „Common Object Request Broker Architecture“ (CORBA) ist ein von der „Object Management Group“ (OMG) festgelegter Standard. Bei der OMG handelt es sich um eine internationale, nicht profitorientierte Organisation, die 1989 gegründet wurde. Ihr gehören mittlerweile über 800 Mitglieder an - darunter Hersteller von Informationssystemen, Softwareentwickler und Anwender [Sie00]. Sie ist bestrebt, den Einsatz und die Verbreitung objektorientierter Technologien bei der Softwareentwicklung zu fördern. Hauptsächliche Ziele der OMG sind

- die Wiederverwendbarkeit
- die langfristige Wartbarkeit
- die Portierbarkeit und
- die Integrationsfähigkeit (Interoperabilität)

von objektbasierter Software. Die OMG definiert Standards und Richtlinien, entwickelt allerdings keine Produkte um selbige zu implementieren, dies ist ausschließlich Aufgabe der Hersteller. Sie möchte einen Rahmen schaffen für die Entwicklung von Anwendungen, die in einer heterogenen, verteilten IT-Umgebung zusammenarbeiten [OMG1]. Als konzeptionelle Infrastruktur dafür wurde von der OMG die „Object Management Architecture“ (OMA) vorgestellt. Die OMA dient als Basis für alle von der OMG entwickelten Standards. Das Referenzmodell der OMA beschreibt eine Softwarelandschaft, in der Anwendungsprogramme kooperieren und dabei auf allgemein verfügbare Dienste unterschiedlichen Abstraktionsniveaus zurückgreifen. Die CORBA-Spezifikation dient als ein erster Schritt, um diese Vision Realität werden zu lassen.

3.1 OMA-Referenzmodell

Das Referenzmodell identifiziert fünf Komponenten sowie die Schnittstellen (Interfaces) und Protokolle, die für den Zugriff auf diese Komponenten benötigt werden, siehe Abbildung 9.

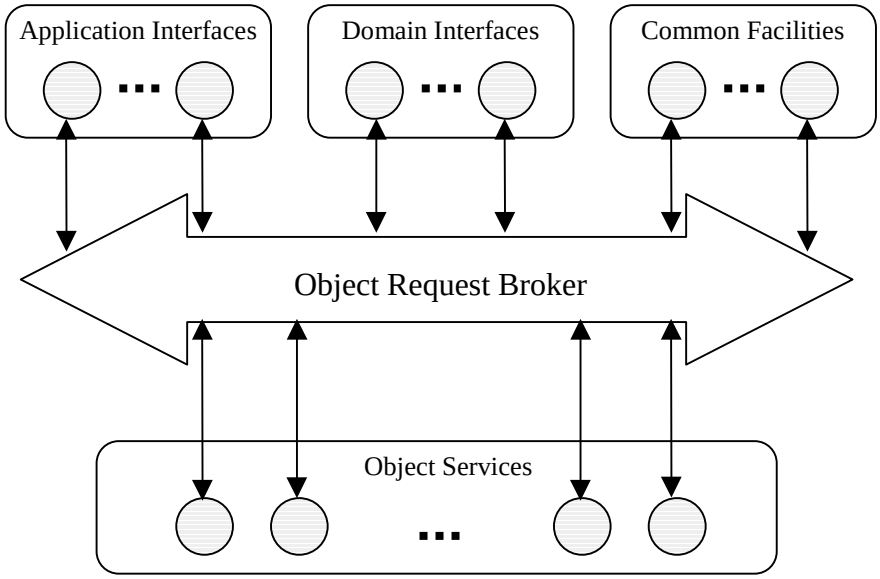

Abbildung 9: schematische Darstellung der „Object Management Architecture“ (nach [OMG2])

Zentraler Bestandteil der OMA ist der „Object Request Broker“ (ORB). Er dient als Infrastruktur, die es allen Objekten in einer verteilten Umgebung ermöglicht, miteinander zu kommunizieren. Dabei bleibt es jedem Objekt verborgen in welcher Programmiersprache sowie auf welcher Hardware- und Betriebssystemplattform das jeweils andere Objekt implementiert wurde und wo sich sich dieses Objekt befindet. Bestehende, nicht objektorientierte Anwendungen können an der OMA teilhaben, indem sie in ein Objekt (Object Wrapper) eingebettet werden, das existierende Funktionen nach außen hin über eine Objektschnittstelle anbietet. Die CORBA-Spezifikation standardisiert die ORB-Komponente der OMA.
Bei den „Object-Services“ handelt es sich um grundlegende Dienste für Objekte in einer verteilten, heterogenen Umgebung. Mehrere solcher Basisdienste wurden bereits als „CORBAServices“ standardisiert [Sei99], wie beispielsweise:

- der „Naming Service“, er ermöglicht es einem Objekt, sich unter einem Namen gegenüber anderen Objekten bekannt zu machen
- der „Event Service“, er dient zum Austausch von Ereignismeldungen
- der „Persistence Service“, er gestattet das dauerhafte Speichern von Objekten
- der „Transaction Service“, er sorgt für die verlässliche, atomare Ausführung von Operationen in Netzen
- der „Security Service“, er unterstützt die Identifikation und Authentisierung von Personen und Objekten sowie Zugriffskontrolle und Verschlüsselung

Leider werden die von der OMG spezifizierten Dienste nur zum Teil von Herstellern implementiert. Der „Naming Service“ und der „Event Service“ haben dabei die

bisher breiteste Unterstützung erfahren. Weniger verbreitet ist hingegen der „Security Service“, dessen Spezifikation zudem nur teilweise implementiert wird [Sie00].
Die OMA-Komponenten „Common Facilities“ und „Domain Interfaces“ stellen eine Zusammenfassung von Diensten auf Anwendungsebene dar. Erstere sind branchenunabhängig. Letztere beziehen sich auf bestimmte Wirtschaftsbereiche, wie Gesundheitswesen, Telekommunikation, Finanzwesen [OMG2]. Druckdienste oder E-Mail-Dienste sind Beispiele für „Common Facilities“, die auch als horizontale „CORBAfacilities“ bezeichnet werden. Die „Domain Interfaces“ werden hingegen auch vertikale „CORBAfacilities“ genannt, beispielsweise wurde für das Anwendungsgebiet Gesundheitswesen (CORBAmed) ein Dienst für die Identifikation von Personen („Person Identification Service“) spezifiziert [Sie00].
Die „Application Interfaces“ repräsentieren innerhalb der OMA die eigentlichen Anwendungen der Nutzer dieser Architektur. Daher werden „Application Interfaces“ auch nicht von der OMG standardisiert.

3.2 CORBA-Objektmodell

Das Objektmodell, welches der CORBA-Spezifikation zugrunde liegt, unterscheidet Clients und Objekte. Ein Objekt ist eine identifizierbare, gekapselte Einheit, die einen oder mehrere Dienste als aufrufbare Methoden (Operationen) anbietet, daher wird ein Objekt gelegentlich auch als Server bezeichnet. Ein Client entspricht einer Einheit, die in der Lage ist, einen Dienst anzufordern. Dabei kann ein Objekt seinerseits auch die Rolle eines Clients gegenüber anderen Objekten einnehmen [OMG1]. Die von einem Objekt angebotenen Dienste werden in Form einer Schnittstellenbeschreibung des Objektes definiert. Diese Schnittstellendefinition (Interface Definition) dient gewissermaßen als ein Vertrag zwischen Client und Objekt, der für alle Clients und Objektimplementierungen festschreibt, welche Operationen zur Verfügung stehen bzw. zur Verfügung zu stellen sind, siehe Abbildung 10.

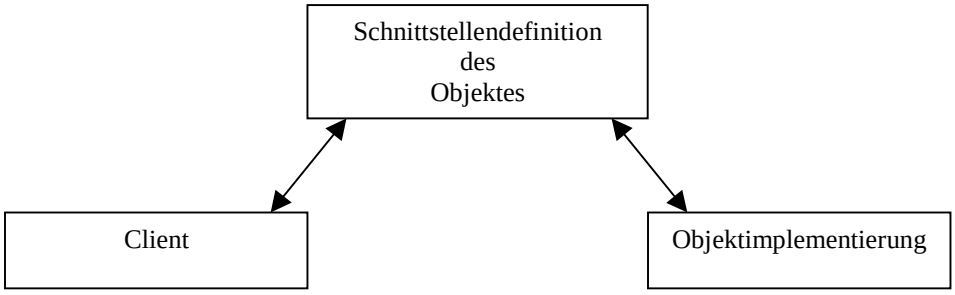

Abbildung 10: schematische Darstellung der Trennung von Schnittstelle und Implementierung eines Objektes

Die Beschreibung der Objektschnittstellen erfolgt in der „Interface Definition Language“ (IDL). Die IDL ist eine von der OMG standardisierte Sprache. Eine Schnittstellenbeschreibung enthält also eine Anzahl von IDL-Methodendefinitionen, die jeweils den Namen der Methode sowie die Art der zugehörigen Aufrufparameter und Rückgabewerte festlegen. Zu beachten ist, dass der CORBA-Standard das Vererbungskonzept ausschließlich auf der Ebene der Objektschnittstellen unterstützt (interface inheritance), nicht jedoch auf der Ebene der Objektimplementierungen (implementation inheritance) [Sie00]. Dies soll an einem Beispiel verdeutlicht werden. Angenommen, eine existierende Schnittstelle *„S1“* enthält eine Methode *„M1“* und eine von *„S1“* abgeleitete Schnittstelle *„S2“* definiert eine Methode *„M2“*. Eine Objektimplementierung zu *„S2“* muss nun sowohl *„M2“* als auch *„M1“* durch entsprechenden Programmquelltext implementieren. (Bei Vererbung auf Implementierungsebene müsste diese Objektimplementierung nur den Programmquelltext für *„M2“* enthalten.)
Die Abbildung der IDL in eine konkrete Programmiersprache, wird „Language-Mapping“ genannt und ebenfalls von der OMG standardisiert. Entsprechende Standards wurden u.a. für die Programmiersprachen Ada, C, C++, COBOL, Java und Smalltalk festgelegt.
Aus Sicht des Clients wird ein Objekt duch eine eindeutige Objektreferenz identifiziert, wobei ein Objekt auch mehrere Objektreferenzen besitzen kann. Um einen Dienst eines Objektes in Anspruch zu nehmen, sendet der Client eine Anforderung (Request) an das Objekt, siehe Abbildung 11.

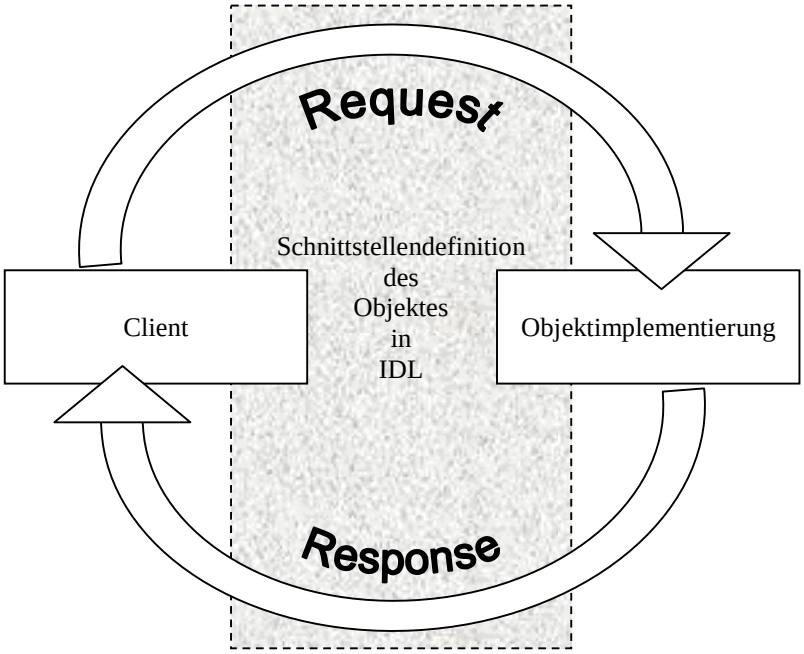

Abbildung 11: schematische Darstellung der Beziehungen zwischen Client und Objektimplementierung

Ein solcher Request entspricht einer Nachricht, die Informationen, wie den Namen des Zielobjektes (die Objektreferenz), den Namen der angeforderten Methode und gegebenenfalls zugehörige Parameter für den Methodenaufruf, enthält. Handelte es sich um eine Methode, die Ergebnisse zurückliefert, so werden diese anschließend als Antwort (Response) an den Client übermittelt. Sind bei der Ausführung der Methode Ausnahmebedingungen (Exceptions) aufgetreten, erhält der Client eine entsprechende Nachricht.

3.3 Aufgaben und Struktur eines ORB

Die IDL-Beschreibung der Objektschnittstelle ermöglicht es, Client und Objektimplementierung durch den ORB so weit voneinander zu entkoppeln, dass die resultierende Kommunikation plattformübergreifend und unabhängig von den eingesetzten Programmiersprachen stattfindet, siehe Abbildung 12.

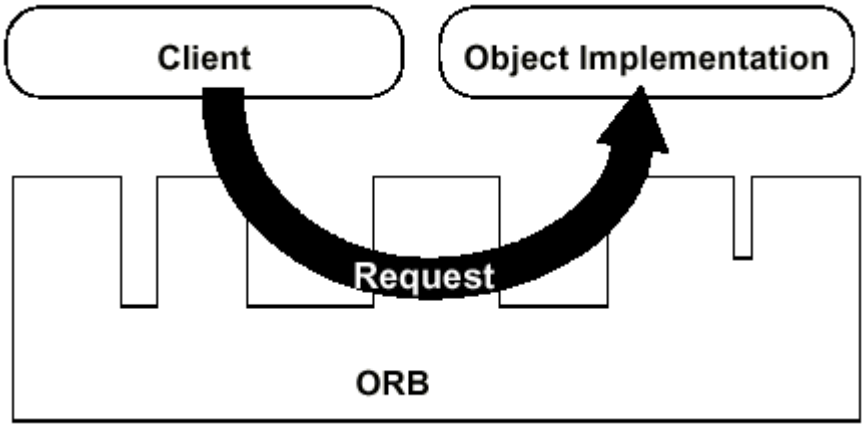

Abbildung 12: schematische Darstellung der Weiterleitung einer Anforderung (Request) durch den ORB [OMG1]

Bei der Weiterleitung eines Requests an ein Objekt, ist der ORB verantwortlich für:

- das Finden der Objektimplementierung
- das Vorbereiten der Objektimplementierung auf den Empfang des Requests
- die Übertragung der Daten, einschließlich
    - der Konvertierung der Parameter der Operation auf der Clientseite und des Zusammenstellens einer Nachricht (Marshaling)
    - des Transportes dieser Nachricht
    - der entsprechenden Konvertierung der Parameter auf Seiten der Objektimplementierung
- den Aufruf der entsprechenden Methode

Die CORBA-Spezifikation legt die Schnittstellen fest, über welche der ORB seine Dienste für Client und Objektimplementierung zugänglich macht, siehe Abbildung 13.

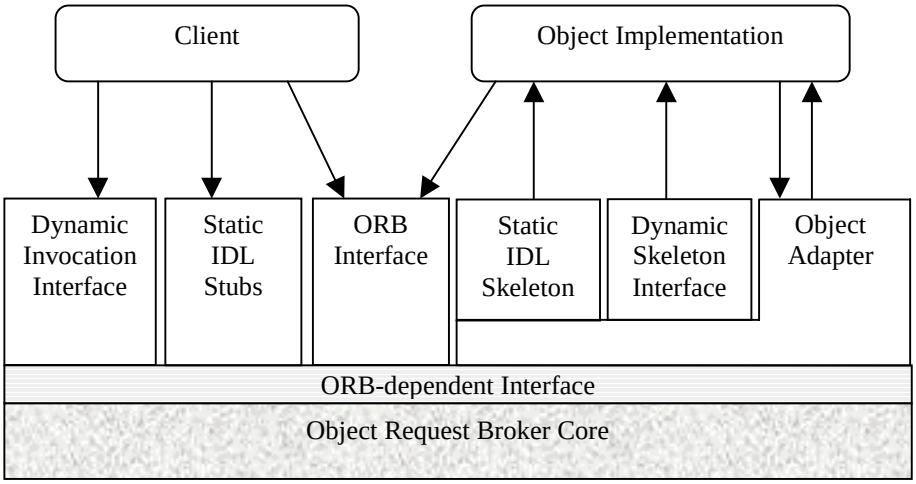

Abbildung 13: schematische Darstellung der Struktur eines ORB (nach [OMG1])

Ein ORB muss nicht als einzelne Komponente implementiert werden, der CORBA-Standard erfordert lediglich die Unterstützung der ORB-Schnittstellen. Eine ORB-Implementierung kann daher aus einer Sammlung von Diensten und Komponenten bestehen. Als ORB-Kern (ORB Core) wird der Teil eines ORB bezeichnet, welcher für die grundlegende Repräsentation von Objekten und die Übertragung der Requests sorgt.
Ein ORB bietet für Clients statische und dynamische Schnittstellen. Das „Common“ in CORBA weist auf diese beiden Arten von Schnittstellen hin [Orf98]. Auf Seiten der Objektimplementierung sind ebenfalls statische und dynamische Schnittstellen vorhanden. Aus Sicht der Objektimplementierung ist es nicht erkennbar, also transparent, ob ein Client für den Methodenaufruf die statische oder die dynamische Schnittstelle zum ORB genutzt hat. Ein Client kann ebenso nicht feststellen, ob eine Objektimplementierung eine statische oder eine dynamische Schnittstelle zum ORB verwendet. Die statischen Schnittstellen (IDL-Stubs und IDL-Skeletons) sind spezifisch für den jeweiligen Objekttyp. Sie werden jeweils durch einen IDL-Compiler aus der IDL-Definition der Objektschnittstelle generiert. Bei der Erzeugung des IDL-Stubs übersetzt der IDL-Compiler in die Programmiersprache, in der der Client erstellt werden soll, wobei das „Language-Mapping“ angewendet wird. Entsprechend wird für das IDL-Skeleton die Programmiersprache der Objektimplementierung verwendet. Ein IDL-Stub wird Teil des Clients und agiert als lokaler Stellvertreter (Proxy) für das Objekt. Damit bleiben für den Client der Ort des Objektes sowie alle Details der Objektimplementierung transparent.

Weil bei Verwendung von IDL-Stub und IDL-Skeleton die IDL-Beschreibung der Objektschnittstelle zur Übersetzungszeit vorliegen muss, werden diese Schnittstellen als statisch bezeichnet.
Definitionen von Objektschnittstellen können außerdem über ein „Interface Repository“ (IR) zur Verfügung gestellt werden. Dieses „Interface Repository“ dient als verteilte Datenbasis, in der Informationen über Definitionen von Objektschnittstellen gespeichert sind [Orf98]. Diese Metadaten können zur Laufzeit nachgeschlagen werden. Das IR besitzt dabei die gleiche Ausdruckskraft für die Beschreibung von Objektschnittstellen wie die statischen IDL-Definitionen. Ein Client kann mit Hilfe des IRs zur Laufzeit die Schnittstellen von Objekten erkunden und so auch Informationen über Objekte erhalten, die zur Übersetzungszeit des Clients noch nicht vorhanden waren. Mit den so gewonnenen Daten kann er dynamisch einen Request zusammenstellen, ein Stub ist dabei nicht notwendig. Der Client nutzt das „Dynamic Invocation Interface“ (DII), um einen solchen Request an ein Objekt zu senden. Diese Schnittstelle ist unabhängig vom Typ des Zielobjektes, das heißt es gibt nur eine Ausprägung dieser Schnittstelle.
Das „Dynamic Skeleton Interface“ (DSI) und das statische IDL-Skeleton sind die Schnittstellen, über welche die Objektimplementierung die Requests empfängt. Das statische IDL-Skeleton bietet – ähnlich einem Stub auf der Clientseite – eine Schnittstelle für jeden Dienst, den ein Objekt anbietet.
Das DSI liefert die Möglichkeit eingehende Methodenaufrufe an Komponenten weiterzuleiten, ohne das ein IDL-Skeleton für diese Komponenten benötigt wird. Das DSI erlaubt es, beliebige Requests entgegenzunehmen. Es ist, analog zum DII, nicht spezifisch an einen Objekttyp gebunden, daher existiert pro ORB-Implementierung nur eine Ausprägung dieser Schnittstelle. Laut [Sie00] wird das DSI u.a. für Brücken zwischen ORBs eingesetzt.
Objektimplementierungen greifen primär über den Objektadapter auf Dienste des ORBs zu. Ein Objektadapter bildet die Laufzeitumgebung für Objekte. Zu seinen Aufgaben gehören u.a. das Aktivieren und Deaktivieren von Objekt-implementierungen, die Generierung und Vergabe der Objektreferenzen, das Weiterleiten der Requests sowie das Registrieren der vorhandenen Objekt-implementierungen in ein „Implementation Repository“. Das „Implementation Repository“ enthält Informationen zu den Objektimplementierungen, mit deren Hilfe ein ORB Objektimplementierungen finden und aktivieren kann. Prinzipiell kann eine ORB-Implementierung unterschiedliche Typen von Objektadaptern umfassen. Eine ORB-Implementierung kann für besondere Anwendungsfälle optimiert sein und dementsprechend spezifische Dienste enthalten, die nur durch einen speziellen Objektadaptertyp für die Objektimplementierung nutzbar gemacht werden können. Die Objektimplementierung wählt sich entspechend den gewünschten Objektadapter aus.

Über das „ORB Interface“ greifen sowohl Client als auch Objektimplementierung auf ORB-Funktionen zu, die unabhängig vom Typ des Objektadapters und vom Objekttyp sind. Dazu gehören beispielsweise eine Funktion, die Objektreferenzen in Strings umwandelt, und eine Funktion für die entgegengesetzte Konvertierung.

3.4 Einige CORBA-Details im Kontext der bisherigen Entwicklung

Die CORBA-Spezifikation wurde und wird seit ihrer ersten Veröffentlichung ständig weiterentwickelt und verfeinert. Die ersten Versionen „CORBA 1.x“ legten die Struktur eines ORBs weitgehend fest. Diverse Details, die bei der Implementierung eines CORBA-konformen ORBs zu lösen waren, wurden vom Standard nicht spezifiziert und galten gewissermaßen als „Übung“ für die Hersteller. Mit „CORBA 1.x“ ließ sich die Interoperabilität von Objekten erreichen. Da das Format der Objektreferenzen vom Standard nicht vorgeschrieben wurde, galt dies nur, solange ein Anwender die CORBA-Implementierung eines Herstellers einsetzte.
Die Version „CORBA 2.0“ erschien 1995 und adressierte hauptsächlich die Zusammenarbeit von ORB-Implementierungen verschiedener Hersteller. Das mit „CORBA 2.0“ vorgestellte „General Inter-ORB Protocol“ (GIOP) definiert die Nachrichtenformate und die Datenrepräsentation für die Kommunikation zwischen ORBs [OMG1]. Insbesondere führt GIOP mit der Definition von „Interoparable Object References“ (IORs) ein einheitliches Format für den Austausch von Objektreferenzen zwischen ORBs ein. GIOP wurde so entworfen, dass es direkt auf einen verbindungsorientierten Transportdienst aufsetzt. Das „Internet Inter-ORB Protocol“ (IIOP) legt fest, wie GIOP-Nachrichten unter Verwendung von TCP/IP-Verbindungen ausgetauscht werden. Darüber hinaus kann die Anbindung von GIOP an weitere Transportprotokolle erfolgen. Neben GIOP kann eine ORB-Implementierung auch Protokolle verwenden, die an spezifische Umgebungen angepasst wurden und daher „Environment Specific Inter-ORB Protocols“ (ESIOPs) genannt werden. So wurde beispielsweise von „CORBA 2.0“ ein ESIOP für DCE (Distributed Computing Environment) angegeben.
Damit eine ORB-Implementierung kompatibel zu „CORBA 2.0“ ist, muss sie IIOP unterstützen. Dies kann auf direktem Wege geschehen, indem der ORB-Kern IIOP verwendet. Falls der ORB-Kern ein anderes Potokoll nutzt, muss eine Brücke (Bridge) von diesem Protokoll nach IIOP bereitgestellt werden. Eine solche Brücke wird auch als „Half-Bridge“ bezeichnet. IIOP dient in diesem Fall als „Backbone-Protokoll“, das gegebenenfalls mehrere „ORB-Domains“ mit jeweils ORB-spezifischem Protokoll via „Half-Bridge“ verbindet, siehe Abbildung 14.

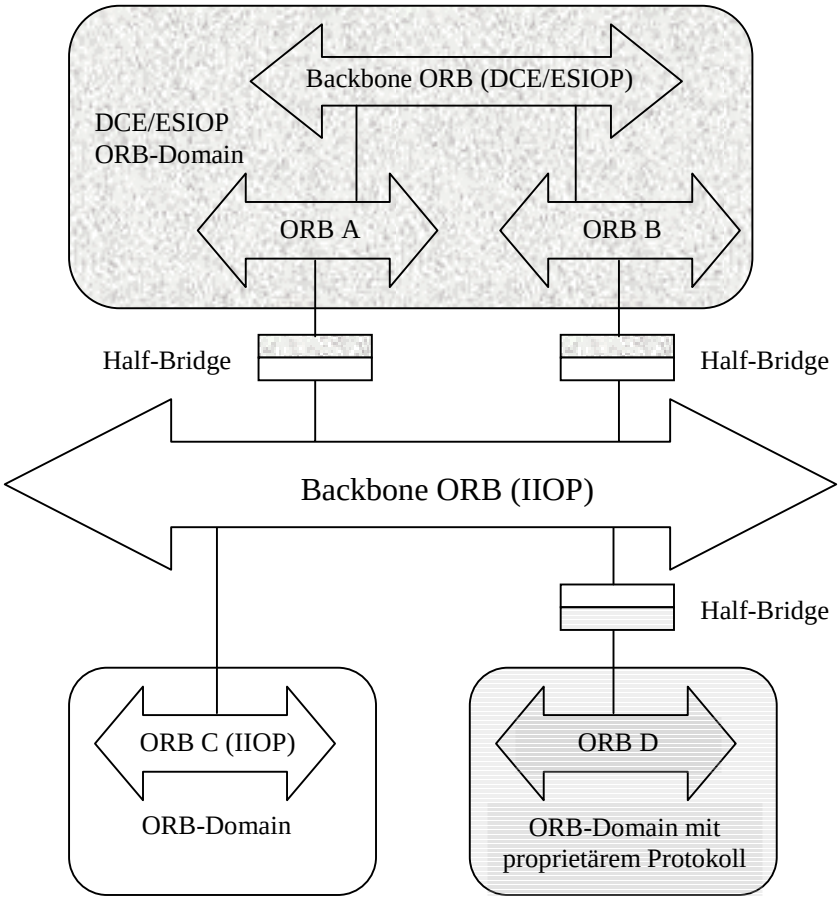

Abbildung 14: schematische Darstellung der Interoperabilität verschiedener ORB-Implementierungen durch IIOP (nach [Orf98] und [OMG1])

Zu einer „ORB-Domain“ können mehrere Komponenten gehören. Eine „ORB-Domain“ ist ein Bereich mit gemeinsamer Charakteristik, die administrative, sicherheitsrelevante oder sonstige technologische Aspekte betreffen kann. Der CORBA-Standard verwendet den Begriff der „Domain“ bewusst in einer sehr allgemeinen Form, um der Vielfalt möglicher Einteilungen gerecht zu werden. Mit „CORBA 2.0“ wurde auch das DSI eingeführt.
Die 1998 vorgestellte Version „CORBA 2.2“ stellte mit dem „Portable Object Adapter“ (POA) einen Objektadapter vor, den künftig jede ORB-Implementierung enthalten sollte [Ave98]. Vor „CORBA 2.2“ galt der „Basic Object Adapter“ (BOA) als ein Objektadapter, der grundlegende Funktionen anbietet und den typischerweise alle ORBs unterstützen sollten. Der BOA war allerdings im Standard nicht ausreichend spezifiziert worden, so dass sämtliche ORB-Implementierungen herstellerspezifische Erweiterungen vornehmen mussten [Orf98]. Entsprechend waren Teile des Programmcodes von Objektimplementierungen auf diese hersteller-spezifischen Erweiterungen angewiesen und meist nicht ohne Änderungen auf

einer ORB-Implementierung eines anderen Herstellers lauffähig. Obwohl der BOA in der CORBA-Spezifikation durch den POA abgelöst wurde, können ORB-Implementierungen den BOA weiterhin (zusätzlich) unterstützen [Sie00]. Neben der Portabilität auf Seiten der Objektimplementierung enthält der POA weitere Verbesserungen und Konfigurationsmöglichkeiten, die sich u.a. auf Skalierbarkeitsaspekte von Objektimplementierungen beziehen.
„CORBA 2.3“ wurde 1999 standardisiert. Mit „CORBA 2.3“ bietet der Standard die Möglichkeit komplexe Datenstrukturen, wie z.B. eine Baumstruktur, als Objekt-Werte „by Value“ zum Client zu übertragen [Sie00]. Der Client arbeitet dann mit einer lokalen Kopie dieser Datenstruktur. Bislang konnte lediglich mit Hilfe von Objektreferenzen, also „by Reference“, auf (gegebenenfalls entfernte) Objekte zugegriffen werden.
Für „CORBA 3“ hat sich die OMG weitere Ergänzungen und Veränderungen des Standards vorgenommen, die zu drei Schwerpunktthemen der Softwareentwicklung gruppiert werden können ([Sie00], [Sta1]):

1. Integration mit dem Internet und mit Java
2. Quality of Service (QoS)
3. Komponenten (CORBA Component Model)

Zur engeren Integration von CORBA und Java wurde ein „Reverse Language-Mapping“, also eine Abbildung von Java nach IDL standardisiert. Einen weitereren Schritt zur Integration von CORBA und dem Internet stellt die CORBA-Firewall-Spezifikation dar. Sie definiert Möglichkeiten, wie Firewalls IIOP-Datenverkehr verarbeiten können [Sta2]. Der „Interoperable Naming Service“ (INS) ergänzt den bisherigen „Naming Service“ durch ein Adressierungsschema, das auf URLs (Uniform Rescource Locators) basiert und damit der Adressierung von Ressourcen im „World Wide Web“ ähnelt [Sta2]. Somit lassen sich Informationen, die bisher in IORs enthalten waren auch als URL-Namen repräsentieren. Allerdings gibt es in diesem Zusammenhang Einschränkungen, da sich nicht alle Informationen, die eine IOR enthalten kann, mit der URL-Syntax ausdrücken lassen [Sei00].
Obwohl CORBA plattformneutral ist, gibt es verschiedene Einsatzszenarien, bei denen es wünschenswert ist, die Merkmale der verwendeten Plattform zu berücksichtigen bzw. Anforderungen an die Plattform spezifizieren zu können. Gilt es, bestimmte Qualitätskriterien bei der Datenübertragung einzuhalten (etwa die „Lebensdauer“ von Requests sowie die maximale Wartezeit auf die entsprechende Antwort) oder das Management einer Warteschlange von Requests zu beeinflussen, so können im Rahmen von „CORBA 3“ Prioritäten und weitere Merkmale des Nachrichtenaustausches in Form von „Policies“ angegeben werden. Die zugehörigen Definitionen werden unter der Bezeichnung „Quality of Service“ in der Spezifikation

zusammengefasst. Außerdem werden die Kommunikationsmöglichkeiten zwischen Client und Objektimplementierung erweitert. Bis einschließlich „CORBA 2.3“ gab es folgende Arten des Methodenaufrufes:

- synchron: Hier blockiert der Client so lange, bis die Objektimplementierung die Methode abgearbeitet hat und die Ergebnisse beim Client eingetroffen sind.
- verzögert synchron (deferred synchronous): Nachdem der Request vom Client gesendet wurde, erhält das Clientprogramm wieder die Kontrolle und kann danach durch entsprechende Funktionsaufrufe feststellen, ob die Ergebnisse eingetroffen sind. Diese Art des Methodenaufrufes ist aber nur für die Benutzung des DII definiert worden.
- „oneway“: Wie der Name bereits andeutet, können „oneway“-Methoden keine Ergebnisse liefern und keine Ausnahmebedingungen definieren. Der CORBA-Standard garantiert allerdings nicht, dass der Request bei der Objekt-implementierung ankommt und die angeforderte Operation ausgeführt wird, sondern spricht von einer „Best Effort“-Semantik, die von der ORB-Implementierung verwendet wird. Da nicht näher angegeben wurde was unter „Best Effort“ zu verstehen ist, blieb die Art und Weise der Implementierung den ORB-Herstellern überlassen.

„CORBA 3“ führt mit „Asynchronous Method Invocation“ (AMI) asynchrone (nicht blockierende) Methodenaufrufe, sowohl für die Nutzung der dynamischen als auch der statischen Schnittstellen eines ORBs, ein. Ein Spezialfall von AMI sind Methodenaufrufe per „Time-Independent Invocation“ (TII). Hier werden Requests nach dem „Store and Forward“-Prinzip, ähnlich wie bei e-mail, transportiert. Das heisst Client und Objektimplementierung müssen nicht zur gleichen Zeit aktiv sein. Das TII ermöglicht somit eine „Stapelverarbeitung“ für Requests und wird z.B. in bestimmten Transaktionsumgebungen genutzt. Die Requests können auf ihrem Weg zur Objektimplementierung gegebenenfalls über mehrere CORBA-Router transportiert und dabei für unbestimmte Zeit auf dem jeweiligen CORBA-Router zwischengespeichert werden. In diesem Zusammenhang wird der CORBA-Standard um das „Interoperable Routing Protocol“ erweitert. Es baut auf GIOP auf und fügt u.a. Wegeinformationen hinzu. Für die Nutzung von CORBA in Systemen mit Echtzeitanforderungen enthält „CORBA 3“ die optionale Erweiterung „Real-Time CORBA“ (RT-CORBA). RT-CORBA ist auf Systemumgebungen spezialisiert, die ein vorhersagbares Leistungsverhalten von Anwendungen (inklusive garantierter Intervallbegrenzungen für Antwort- bzw. Verzögerungszeiten und entsprechend gesteuerter Ressourcennutzung) erfordern. Für den Einsatz von CORBA auf Systemen, die nur über sehr begrenzte Ressourcen verfügen (z.B. auf portablen Geräten wie Mobiltelefon oder PDA (Personal Digital Assistant)), wurde eine reduzierte Version von CORBA festgelegt, die „minimumCORBA“ genannt wird.

Sie verzichtet beispielsweise auf die dynamischen Schnittstellen eines ORBs [OMG1]. Eine Verbesserung der Fehlertoleranz CORBA-basierter Systeme soll ab „CORBA 3“ die Spezifikation von „Fault-Tolerant CORBA“ ermöglichen. „Fault-Tolerant CORBA“ kann durch Replikationsmechanismen dafür sorgen, dass mehrere identische Objekte zu einer Objektgruppe zusammengefasst werden [Sta2]. Eine solche Objektgruppe erscheint gegenüber dem Client als einzelnes Objekt. Dies erhöht die Zuverlässigkeit der betreffenden Anwendung, die auch bei Ausfall eines Objektes der Objektgruppe noch verfügbar ist.

Das „CORBA Component Model“ (CCM) ergänzt das bisherige Objektmodell von CORBA durch die Einführung von „Komponenten" (Components) [Sie00]. Das CCM soll u.a. die Programmentwicklung auf der Serverseite von CORBA vereinfachen. Komponenten sind serverseitige Objekte. Sie können mehrere Schnittstellen besitzen, wobei es keine Vererbungsbeziehungen zwischen diesen Schnittstellen geben muss. Anwendungen sollen typischerweise aus mehreren Komponenten bausteinartig zusammengesetzt werden. Dabei wird die Wiederverwendbarkeit von Komponenten angestrebt. Die Komponenten werden in einem „Container“ installiert. Der Container stellt eine standardisierte Softwareumgebung für die Ausführung der Komponenten dar. Er enthält einen ORB samt POA und bietet standardisierte Schnittstellen zu Diensten. So stellt der Container u.a. Transaktions-, Ereignis-, Sicherheits- und Persistenzdienste – aus Sicht der Programmierung von Komonenten – auf einem höheren Niveau bereit, als dies bei den bisherigen CORBA-Services der Fall war [Sie00]. Darüber hinaus wird im Rahmen von CCM auch auf Fragen der Softwareverteilung und -installation eingegangen. In diesem Zusammenhang legt das CCM ein plattformübergreifendes (Paket-)Format für die Auslieferung von Komponenten fest, das u.a. Konfigurationsdateien mit plattformneutralen XML-Beschreibungen (Extensible Markup Language) enthält.

Die CCM-Spezifikation orientiert sich sehr eng an den Festlegungen zu „Enterprise JavaBeans“ (EJB), einer Spezifikation der Firma „Sun Microsystems“. „CORBA-Components“ können daher mit „Enterprise JavaBeans“ zusammenarbeiten. Im Gegensatz zu EJB sind Komponenten bei CCM nicht an eine bestimmte Programmiersprache gebunden.

„CORBA 3“ umfasst also ein breites Spektrum an Erweiterungen des bisherigen CORBA-Standards, an denen allerdings zum Teil noch gearbeitet wird. Daher erschien im Jahr 2000 der „CORBA 2.4“-Standard, in dem bereits ein Teil der Neuerungen, wie beispielsweise INS, AMI, RT-CORBA, aufgenommen bzw. berücksichtigt wurde [OMG1].

4 Tivoli TME 10

4.1 Überblick

„TME 10“ lautet der Name einer Produktfamilie, unter dem der Hersteller „Tivoli Systems“ (kurz: Tivoli) Software-Produkte für das NSM anbietet [Tiv2]. TME steht dabei für „Tivoli Management Environment“. In diesem Kapitel werden Informationen und Zusammenhänge dargestellt, die generell für TME 10 gelten[1]. Die Management-Softwaresuite TME 10 ist besonders auf das NSM von umfangreichen, heterogenen IT-Landschaften (mit vielen beteiligten Systemen) – wie sie typischerweise in großen Unternehmen zu finden sind – ausgerichtet. TME 10 besteht aus dem „Tivoli Management Framework“ (TMF) und einzelnen Managementapplikationen, siehe Abbildung 15.

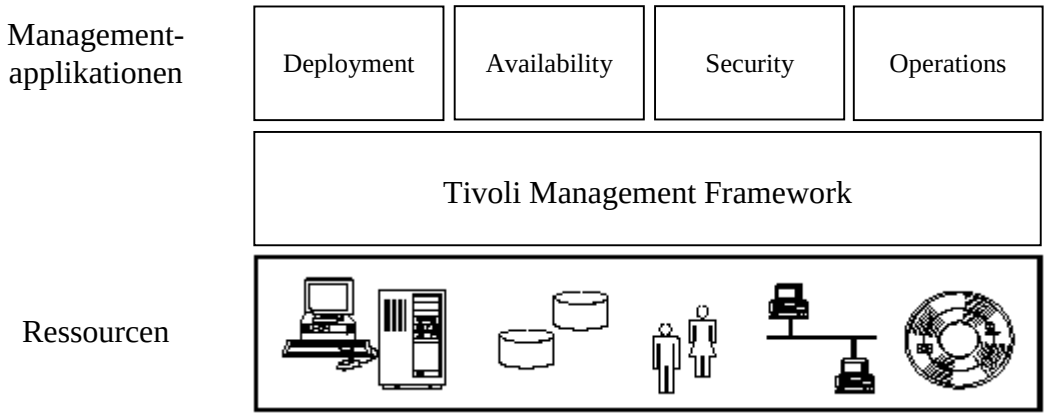

Abbildung 15: Überblick zur „Framework-Architektur“ von Tivoli’s Managementumgebung TME 10 (nach [Tiv1])

Beim TMF handelt es sich um eine objektorientierte, CORBA-basierte Infrastruktur, die grundlegende Dienste und Einrichtungen für die Managementapplikationen bereitstellt. Es bildet eine Abstraktionsschicht zwischen den Ressourcen (IT-Komponenten), die verwaltet werden sollen, und den Managementapplikationen. Dabei werden bestehende Managementmechanismen, die beispielsweise in Betriebssystemen enthalten sind, im Allgemeinen nicht ersetzt, sondern integriert und gegebenenfalls ergänzt. Das TMF umfasst insbesondere:

- eine interne, objektorientierte, verteilte Datenbank (DB), in der Managementdaten gespeichert werden

[1] Die verwendete Literatur bezieht sich im Wesentlichen auf die Versionen 3.2 und 3.6 von „TME 10“.

- einen „CORBA 1.1“ konformen ORB ([Tiv3], [Bor99]), der um zusätzliche Dienste erweitert wurde (und der in der Literatur meist als „Tivoli Object Dispatcher“ oder – unter Verwendung des Programmnamens – als „oserv“ bezeichnet wird)
- die grafische Bedieneroberfläche (GUI – Graphical User Interface), die „TME 10 Desktop“ genannt wird, und die Kommandozeile (CLI – Command Line Interface); beide bieten den Zugriff auf die Managementapplikationen
- eine Konfigurationsoption (kurze Bezeichnung: „RIM“; ausführliche Bezeichnung: „Relational Database Management System Interface Module“) für die Anbindung einer externen relationalen Datenbank, die von einigen Managementapplikationen benötigt wird
- eine einheitliche Installationskomponente für die Installation der Managementapplikationen

Tivoli ordnet die Managementapplikationen der „TME 10“-Reihe im Wesentlichen vier Kategorien zu [Tiv4], siehe Tabelle 3.

Kategorie	zugehörige Produkte sind z.B.	Bedeutung
Deployment Management	TME 10 Software Distribution, TME 10 Inventory	Konfigurationsmanagement: Software-Verteilung Inventarisierung (Dokumentation vorhandener Hard- und Software)
Availability Management	TME 10 Enterprise Console (TEC), TME 10 Distributed Monitoring, TME 10 NetView, TME 10 Performance Management	Fehlermanagement und Leistungsmanagement: Ereignisverarbeitung Überwachung SNMP-Management
Security Management	TME 10 User Administration, TME 10 Security Management (TSecMan)	Sicherheitsmanagement: Benutzerverwaltung Zugriffsschutz durch rollenbasierte Zugriffskontrolle
Operations and Administration	TME 10 Remote Control, TME 10 ADSM (ADSTAR Distributed Storage Manager)	verschiedene Werkzeuge zur Unterstützung und Automatisierung von alltäglichen Administrationsaufgaben: Fernsteuerung Datensicherung

Tabelle 3: Einteilung der TME 10 Managementapplikationen in Kategorien

Über so genannte „Tivoli/Plus“-Module können Managementanwendungen anderer Hersteller in eine Tivoli Frameworkumgebung integriert werden [Tiv5]. Der damit erzielte Integrationsgrad reicht von der Möglichkeit, die betreffende Anwendung vom „Tivoli Desktop“ aus aufrufen zu können, bis zu einer engen Integration mit dem TMF und anderen Managementapplikationen – etwa durch den gemeinsamen Zugriff auf Managementdaten unter Nutzung der internen Datenbank des TMFs.

4.2 Tivoli Management Region

Tivoli verwaltet die IT eines Unternehmens, indem es (mindestens) eine „Tivoli Management Region“ (TMR) etabliert. Zu einer TMR gehören ein TMR-Server, der zuweilen auch als Framework-Server oder Regionserver bezeichnet wird, sowie eine Anzahl von zu verwaltenden Systemen, den TMR-Clients [Tiv6]. Die TMR wird durch die Installation der TMF-Software auf dem TMR-Server eingerichtet. Ein Rechner auf dem die TMF-Software installiert ist, wird (im Kontext) von Tivoli als „Managed Node“ bezeichnet. Der TMR-Server ist also nach der Installation des TMFs der erste Managed Node innerhalb der TMR. In die TMR können nun TMR-Clients aufgenommen werden. Die TMR ist im Allgemeinen durch eine 3-Ebenen-Architektur hierarchisch strukturiert. Der TMR-Server nimmt dabei die obere Hierarchie-Ebene ein, siehe Abbildung 16.

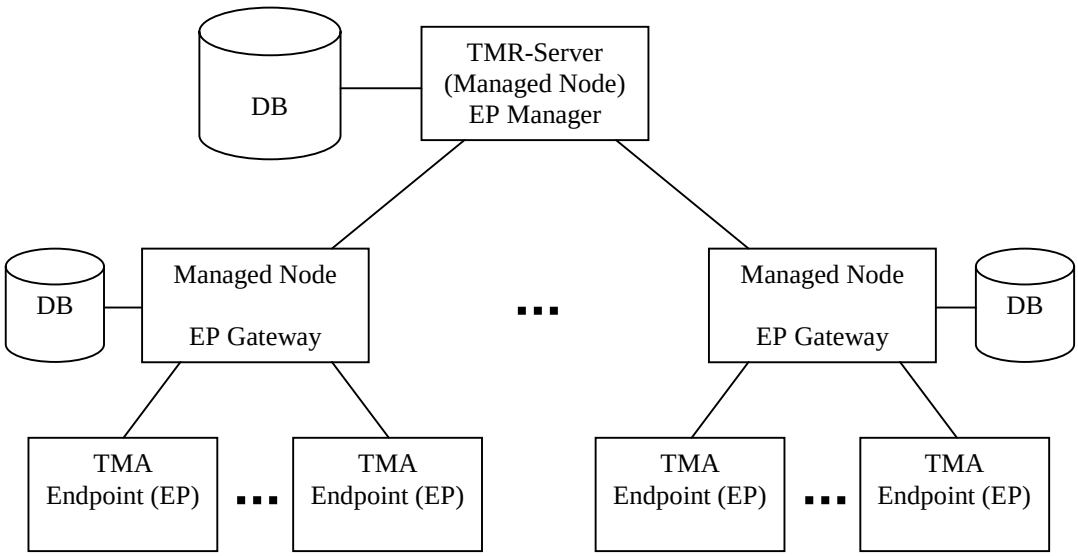

Abbildung 16: schematische Darstellung der 3-Ebenen-Architektur einer TMR (nach [Tiv2] und [Tiv3])

In der unteren Ebene der TMR sind all jene TMR-Clients angesiedelt, die ihrerseits nicht am Management anderer TMR-Clients beteiligt sind. Auf diesen Systemen

ist jeweils eine Software namens „Tivoli Management Agent“ (TMA) installiert. Sie werden deshalb als „TMA-Endpoint“, „Endpoint“ (EP) oder „Managed-Only System“ bezeichnet. Der TMA weist nur einen geringen Ressourcenbedarf auf (im Vergleich zur TMF-Software auf einem Managed Node), worauf auch sein Programmname „lcfd“ (Lightweight Client Framework Daemon) hindeutet. Hier wirkt sich gewissermaßen das „Fundamental Axiom“ (Abschnitt 2.4.2) auf TMA-Endpoints aus. Der TMA basiert ebenfalls auf CORBA, verfügt aber – gegenüber dem „oserv“ auf einem Managed Node – nur über einen limitierten Funktionsumfang [Tiv2], u.a. weil auf TMA-Endpoints keine (TMF-)Datenbank eingerichtet wird. Methoden, die auf einem TMA-Endpoint – im Auftrag von Managementapplikationen – ausgeführt werden sollen, werden zunächst automatisch auf den TMA-Endpoint heruntergeladen und dann vom TMA ausgeführt. Dabei werden die Methoden in einem Zwischenspeicher (Cache) auf der Festplatte des TMA-Endpoints gespeichert. Häufig ausgeführte Methoden müssen daher nur einmal auf den TMA-Endpoint geladen werden. Die einzige Verbindung eines TMA-Endpoints mit der TMR verläuft über ein „Endpoint Gateway“ („EP Gateway“).

Ein „EP Gateway“ ist eine spezielle Software, die auf einem Managed Node installiert werden kann und als Verbindungsglied zwischen einer Menge von TMA-Endpoints und dem Rest der TMR dient. Die Bezeichnung „EP Gateway“ wird sowohl für die entsprechende Software als auch für den betreffenden Managed Node (der in diesem Fall in der Rolle eines „EP Gateways“ fungiert) verwendet. In einer TMR können mehrere „EP Gateways“ eingerichtet werden, die gemeinsam die mittlere Ebene der TMR bilden. Sie verfügen jeweils über eine komplette Installation der TMF-Software (einschließlich „oserv“, Datenbank etc.). In ihrer Datenbank werden allerdings nur Managementinformationen von lokalem Interesse, die also das „EP Gateway“ selbst oder die zugehörigen TMA-Endpoints betreffen, gespeichert. Der TMR-Server hingegen hat die Kontrolle über die gesamte Datenbank in der TMR. Bei der Einrichtung des TMR-Servers wird automatisch auch der „EP Manager“ (als ein Bestandteil der installierten Software) eingerichtet. Der „EP Manager“ verwaltet eine Liste, die alle TMA-Endpoints und „EP Gateways“ der TMR sowie die Zuordnungen von „EP Gateways“ zu TMA-Endpoints enthält. Eine weitere Funktion der „EP Gateways“ besteht darin, dass sie bei der Verteilung von Daten an TMA-Endpoints als Demultiplexer wirken. Soll beispielsweise eine Datei vom TMR-Server an alle TMA-Endpoints verteilt werden, wird diese Datei nur einmal zu jedem „EP Gateway“ gesendet und dann vom „EP Gateway“ aus zu den TMA-Endpoints transportiert.

Das TMF enthält verschiedene Bestandteile, die für die Sicherheit innerhalb einer TMR relevant sind. Es bietet eine integrierte Verwaltungsoption für Tivoli-Administratorkonten. Alle Administratoren, die mit Tivolis Frameworkumgebung arbeiten sollen, müssen als Tivoli-Administratoren eingerichtet werden.

Bei der Installation des TMR-Servers wird ein Root-Administrator, das erste Administratorkonto, für die TMR angelegt. Der Root-Administrator kann weitere Administratorkonten anlegen [Tiv6]. Zur Autorisation von Administratoren stellt das TMF eine Reihe von Privilegien bereit, die den Administratoren zugewiesen werden können. Entsprechend dieser Berechtigungen können Administratoren dann Aufgaben bei der Verwaltung der gesamten TMR oder einzelner Teilbereiche übernehmen.
Bei der Kommunikation zwischen Managed Nodes spielt der TMR-Server eine zentrale, koordinierende Rolle. Er ist insbesondere bei Methodenaufrufen (CORBA-Requests) für das Finden der entsprechenden Implementierung verantwortlich. Zuvor überprüft er, ob der jeweilige Administrator zum Aufruf der Methode berechtigt ist. Die Kommunikation zwischen den Systemen innerhalb einer TMR setzt das Vorhandensein von TCP/IP-Verbindungen voraus. Dies gilt sowohl für den „oserv“ [Tiv3] als auch für den TMA [Tiv7]. Zur Absicherung der Kommunikation kommen Verschlüsselungs- und Prüfsummenverfahren zum Einsatz ([Tiv1], [Tiv2]). Falls erwünscht, kann auf die Verschlüsselung verzichtet werden, dies gilt dann allerdings für die gesamte TMR. Außerdem stehen im TMF weitere Kommunikationsdienste wie Transaktionen oder der „Inter-Object Message“-Dienst (IOM) zur Verfügung. IOM bietet Anwendungen eine asynchrone, bidirektionale, verbindungsorientierte Kommunikation, um Datenmengen größer als 16 Kilobyte effizient zu übertragen.
Eine TMR kann maximal etwa bis zu 200 Managed Nodes (und damit nicht mehr als 200 „EP Gateways“) aufnehmen. Ein „EP Gateway“ kann maximal bis ca. 1000 TMA-Endpoints verwalten. Diese Zahlen sind Abschätzungen, die in der Literatur angegeben werden. Sie beruhen auf prinzipiellen Betrachtungen der Leistungsfähigkeit (Dreh- und Angelpunkt dabei ist die Datenbank, die ja über die Managed Nodes verteilt ist). Sie sind **nicht** so zu verstehen, dass eine TMR 200000 TMA-Endpoints verwalten kann! Die Anzahl der Systeme, die innerhalb einer TMR verwaltet werden können, hängt von vielen Einflussgrößen, und damit letztendlich vom konkreten Anwendungsfall ab. Dabei müssen u.a. folgende Faktoren berücksichtigt werden:

- welche Anforderungen ergeben sich aus der Organisationsstruktur des Unternehmens (z.B. sollen Standorte und Abteilungen zentral oder dezentral verwaltet werden?)
- welche Anforderungen ergeben sich aus der vorliegenden Netzwerktopologie
- welche Anforderungen ergeben sich aus der Sicherheitspolitik (z.B. verträgt sich diese mit der Tatsache, dass es einen Root-Administrator für die TMR gibt?)
- welche Anforderungen sind an die Verfügbarkeit des Managements für die TMR zu stellen (Wenn der TMR-Server nicht verfügbar ist, ist davon die gesamte TMR betroffen, d.h. das Management ist für sämtliche TMR-Clients nicht

verfügbar [Tiv8].)
- welche Managementapplikationen werden betrieben
- welche Anwendungen werden auf den zu verwaltenden Systemen betrieben
- welche Dienstgüte wird von den Managementapplikationen und allen anderen Applikationen erwartet (welche Antwortzeiten sind akzeptabel; wie lange dauert ein Backup der Datenbank etc.)

Die angegebenen Kriterien können dazu führen, in einem Unternehmen mehrere TMRs zu konfigurieren. Dabei können die TMRs – und dadurch die in diesen TMRs installierten Managementapplikationen – über speziell einzurichtende Verbindungen kooperieren oder separat betrieben werden. Prinzipiell besteht die Möglichkeit jeweils zwei TMRs – genauer gesagt die TMR-Server – durch eine „one-way"- oder eine „two-way"-Verbindung zusammenzuschließen, was in dem Austausch von Managementinformationen und der Weiterleitung von CORBA-Requests resultiert. Dadurch erhält entweder nur eine TMR Zugriff auf Managementinformationen in der anderen TMR („one-way") oder beide TMRs tauschen Managementinformationen aus („two-way"). Managementapplikationen in einer TMR können also Ressourcen in einer verbundenen TMR verwalten.

4.3 Profilbasierte Ressourcenverwaltung

In der Datenbank des TMFs sind managementrelevante Daten als Objekte gespeichert. Diese Objekte können reale IT-Ressourcen repräsentieren oder anderweitigen Verwaltungszwecken dienen. Die Datenbank enthält u.a. ein besonders Objekt namens „Tivoli Name Registry" (TNR). Das TNR-Objekt dient als ein Verzeichnis, in dem Name, ID (Identifikator) und Ort von Objekten einer TMR registriert sind. Es ist außerdem der Ort, an dem Referenzen auf Objekte in verbundenen TMRs gespeichert werden. (Nicht alle Objekttypen, die in der DB enthalten sind, können über TMR-Verbindungen ausgetauscht werden [Tiv8].)
Beim Management der Ressourcen wird zwischen ressourcenbezogenen Objekten und Regeln, die auf die Ressourcen angewendet werden sollen, unterschieden. Ressourcenbezogene Objekte, auf die Regeln angewendet werden sollen, heißen „Managed Resources". Die anzuwendenden Regeln werden als „Policy" bezeichnet. Das Aufstellen von „Policies" obliegt den Administratoren. Eine solche Policy kann entweder Vorgabewerte für neu angelegte Ressourcenobjekte enthalten (Default Policy) oder festgelegte Richtlinien für zulässige Werte definieren (Validation Policy). Beispielsweise könnte im Rahmen der Benutzerverwaltung eine Policy festlegen, dass Benutzerkennungen eine maximale Länge von 8 Zeichen besitzen sollen. Es gibt unterschiedliche Typen von „Managed Resources" (Ressourcenarten).

Einige, beispielsweise der Typ für Managed Nodes, werden bereits mit dem TMF installiert. Andere werden erst bei der Installation von Managementapplikationen zur Datenbank hinzugefügt. Eine „Policy Region“ dient als Container für Managed Resources, auf die gemeinsame Regeln (Policies) angewandt werden sollen. Die Managed Resources lassen sich also logisch in Policy Regions gruppieren, wobei eine Managed Resource stets nur Mitglied in einer Policy Region sein kann. Administratoren werden Berechtigungen pro Policy Region erteilt, damit müssen nicht alle Administratoren Berechtigungen für die gesamte TMR erhalten.
Die meisten Managementapplikationen der „TME 10“-Reihe basieren auf Profilen. Es gibt – abhängig von den Managementapplikationen – unterschiedliche Profilarten wie etwa Profile für Benutzerinformationen, Profile für Benutzergruppen, Profile zur Softwareverteilung. Ein Profil enthält ressourcenbezogene Konfigurationsinformationen. Diese Informationen werden in der Datenbank in einem plattformunabhängigen Format gespeichert. Profile können nur in einem Profilmanager angelegt werden. Die Profilmanager dienen als Container für Profile. Profile und Profilmanager sind Managed Resources. Die Profilmanager werden innerhalb von Policy Regions erzeugt, siehe Abbildung 17.

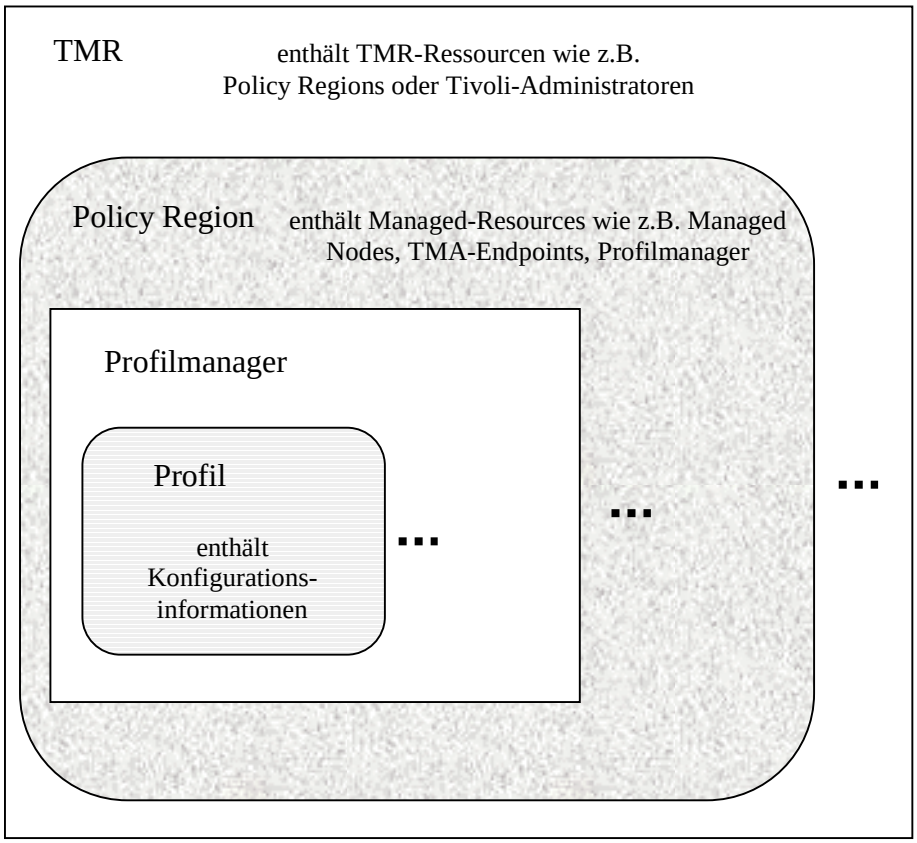

Abbildung 17: schematische Darstellung der Hierarchie bei der Ressourcenverwaltung (nach [Tiv1], [Tiv7], [Tiv8])

Im Profilmanager werden Profile mit einer Menge von Ressourcen, den so genannten „Subscribers“ (Abbonenten), zusammengeführt. Subscribers sind Empfänger von Profilen. Diese Empfänger können die in den empfangenen Profilen enthaltenen Informationen an weitere Empfänger verteilen oder selbst verarbeiten. Empfänger, die diese Informationen selbst verarbeiten und dabei entsprechende Managementoperationen ausführen, werden auch als „Profile Endpoints“ (Profilendpunkte) bezeichnet [Tiv4]. Subscribers sind nicht auf einen Profilmanager beschränkt. Subscribers können Profilmanager, Managed Nodes oder TMA-Endpoints [Tiv7] sein. Damit lassen sich für das Verteilen von Profilen baumartige Verbindungsstrukturen zwischen einem Profilmanager, der die betreffenden Profile enthält, und den Empfängern definieren, siehe Abbildung 18.

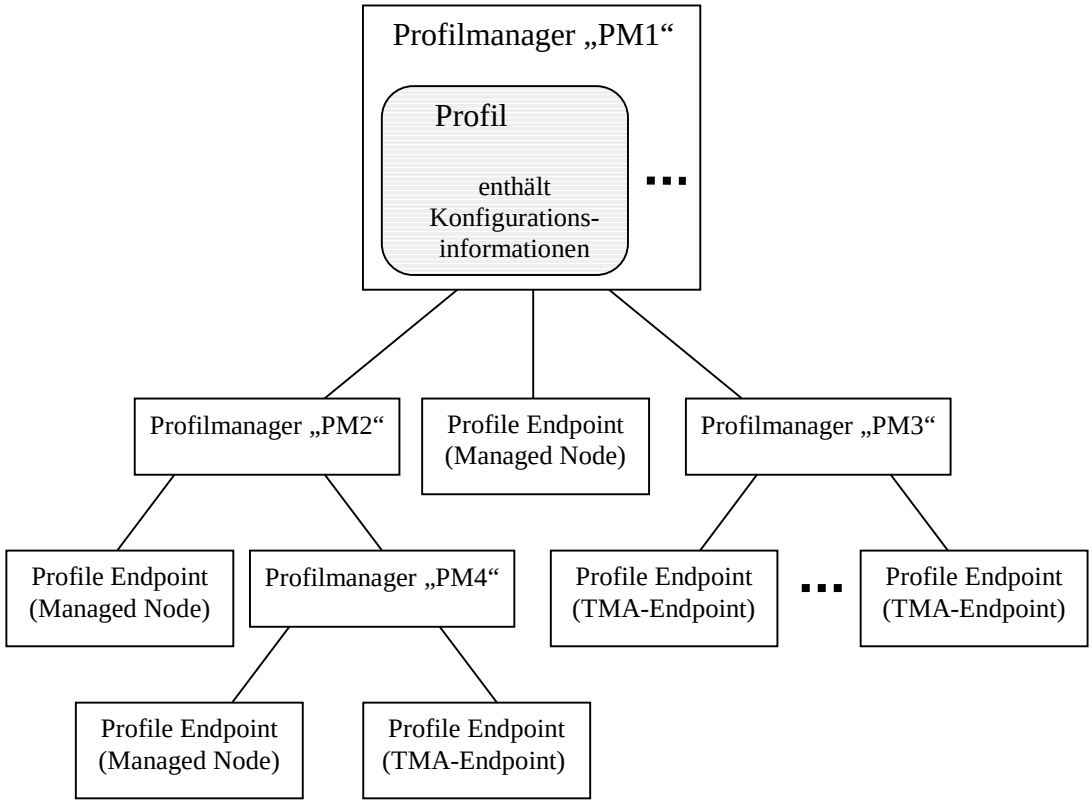

Abbildung 18: grafische Repräsentation eines Beispiels für eine Hierarchie von Profilmanagern (nach [Tiv4], [Tiv7])

Profilmanager, die dabei zu den Subscribers gehören, können als reine Empfängerlisten dienen, das heißt sie müssen nicht notwendigerweise auch selbst Profile enthalten. Profile sind den jeweiligen Managementapplikationen zugeordnet. Sie können mit diesen Applikationen bearbeitet werden. Generell ergibt sich für die profilbasierte Ressourcenverwaltung mit Tivolis Managementapplikationen folgender Ablauf:

1.) Profil anlegen
2.) Profil mit Informationen (über vorhandene Konfigurationseinstellungen der verwalteten Systeme) füllen
3.) Profil bearbeiten
4.) Profil verteilen (und anwenden)

Um ein Profil mit Informationen zu füllen gibt es prinzipiell zwei Möglichkeiten. Zum einen kann der Administrator, unter Benutzung von GUI oder CLI, die Informationen per Hand eingeben. Die andere Möglichkeit besteht darin, mit Hilfe des so genannten „Populating“[2] die gewünschten Informationen über ausgewählte verwaltete Systeme (halb)automatisch abfragen und in einem Profil speichern zu lassen. Als geeignete Informationsquellen kommen Systeme in Frage, die prinzipiell auch als Subscribers dienen könnten. Dabei werden Systemdateien oder -datenbanken dieser Systeme ausgewertet. Das Populating kann ebenfalls unter Benutzung von GUI oder CLI erfolgen. Es stellt also einen teilweise automatisierten Weg zur Informationsgewinnung über Konfigurationseinstellungen der verwalteten Systeme dar. Allerdings lassen sich noch nicht alle für das NSM relevanten Informationen mittels Populating automatisch ermitteln.
Nachdem ein Profil angelegt wurde und Konfigurationsinformationen enthält, kann es weiter editiert oder kopiert werden, ohne dass dies Auswirkungen auf die aktuelle Konfiguration der verwalteten Systeme hat. Das Profil dient also zur Modellierung der vorhandenen bzw. gewünschten Konfiguration dieser Systeme. Wenn das Profil den Wünschen des Administrators entsprechend angepasst wurde, kann es verteilt werden. Dieser Vorgang wird auch als „Distributing“ bezeichnet. Der Administrator kann auswählen, ob das Profil nur an seine direkten Subscribers (in der nächsten Hierarchiestufe) oder an sämtliche Empfänger in allen Ebenen der betreffenden Hierarchie von Profilmanagern verteilt werden soll. Erst wenn ein Profil an einen Profilendpunkt verteilt wird, wirkt es auf die Konfiguration dieses Profilendpunktes. Dabei werden die Profilinformationen mit plattformspezifischen Managementoperationen umgesetzt, wobei der Administrator festlegen kann, ob die Konfiguration des Profilendpunktes ergänzt oder ersetzt werden soll.

[2] „To populate“ ist der englische Ausdruck für „bevölkern“. Gewissermaßen wird durch das Populating ein (eventuell leeres) Profil – im Sinne des NSMs – „mit Leben gefüllt“.

5 Windows NT

Wenn hier von Windows NT die Rede ist, so ist damit das Betriebssystem „Windows NT 4.0“ der Firma Microsoft gemeint. Es handelt sich dabei um ein multitaskingfähiges 32-Bit-Betriebssystem. Windows NT verfügt über integrierte Netzwerkfunktionalität und präsentiert sich gegenüber den Benutzern mit einer grafischen Oberfläche. Windows NT gibt es sowohl in einer Workstation- als auch in einer Server-Ausgabe, wobei sich der grundsätzliche Aufbau beider Varianten nicht unterscheidet. Unterschiede sind hauptsächlich auf dem Gebiet des Einsatzes im Netzwerk vorhanden. Bei der Entwicklung von Windows NT standen folgende Aspekte im Vordergrund:

- Leistungsfähigkeit
- Zuverlässigkeit und Sicherheit
- grafische Oberfläche
- Erweiterbarkeit
- Skalierbarkeit
- Kompatibilität
- Plattformunabhängigkeit

5.1 Architektur

Windows NT besitzt eine modulare Architektur. Beim internen Aufbau von Windows NT wurden Konzepte des Schichtenmodells und des Client/Server-Modells umgesetzt. Nach dem Schichtenmodell wird ein System in eine Anzahl übereinanderliegender funktionaler Schichten eingeteilt. Dabei ist innerhalb der jeweiligen Schicht eine festgelegte Funktionalität zu erbringen, die der übergeordneten Schicht zur Verfügung gestellt wird. Von einer bestimmten Schicht aus kann jeweils nur auf Dienstleistungen der darunterliegenden Schicht zugegriffen werden. Entsprechend dem Client-Server-Konzept sind Bestandteile des Betriebssystems als Serverkomponenten ausgeführt, die gewisse Dienstleistungen anbieten, welche von Clientkomponenten in Anspruch genommen werden können. Die Kommunikation zwischen den Komponenten erfolgt durch den Austausch von Nachrichten.
Windows NT besitzt zwei Betriebsarten, einen privilegierten Modus (Kernel-Modus) und einen unprivilegierten Modus (User-Modus bzw. Benutzermodus). Im Kernel-Modus werden Teile des Betriebssystems ausgeführt, während Anwendungsprogramme und Bestandteile aus höheren Schichten des Betriebssystems im User-Modus ablaufen. Diese Modi korrespondieren mit Betriebsarten des Prozessors auf dem das Betriebssystem ausgeführt wird, im privilegierten Modus steht der

gesamte Satz an Maschinenbefehlen sowie der Zugriff auf alle Speicherbereiche zur Verfügung. In der unprivilegierten Betriebsart sind nur bestimmte Maschineninstruktionen gestattet, der Zugriff auf vom Betriebssystem belegten Speicher ist nicht möglich.
Eine schematische Darstellung der Architektur von Windows NT ist in Abbildung 19 dargestellt.

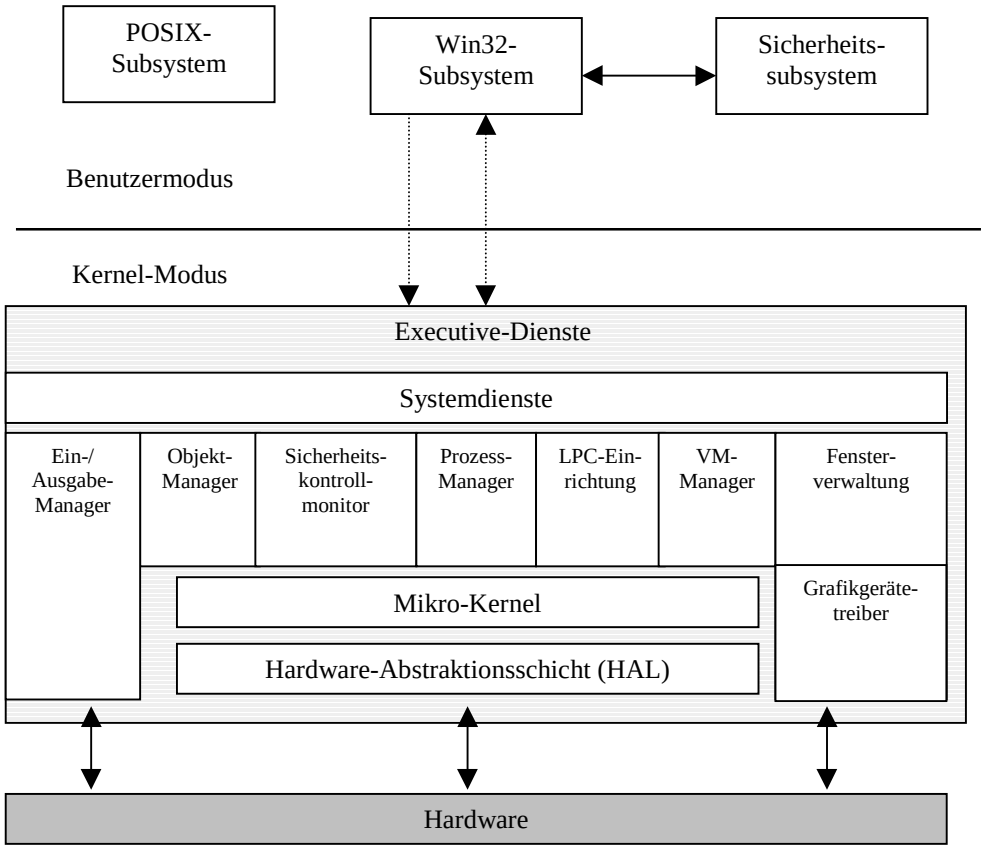

Abbildung 19: modulare Architektur von Windows NT (nach [Zen97])

Der Begriff „Executive-Dienste“ dient zur Bezeichnung sämtlicher Bestandteile von Windows NT, die im Kernel-Modus ausgeführt werden. Die in Abbildung 19 dargestellten Komponenten der Executive-Dienste verdeutlichen den modularen Aufbau von Windows NT. Von diesen Komponenten sind im weiteren Verlauf der vorliegenden Arbeit insbesondere der „Objekt-Manager“ und der „Sicherheits-kontrollmonitor“ von Interesse.
Zu den Bestandteilen von Windows NT, die im unprivilegierten Modus ausgeführt werden, gehören u.a. sog. „Subsysteme“. Dabei handelt es sich um Server-komponenten, von denen zwei Arten unterschieden werden. „Integrale Subsysteme“ erbringen Leistungen, die für das gesamte Betriebssystem wichtig sind, das „Sicherheitssubsystem“ gehört beispielsweise zu dieser Kategorie [MSD98].

„Umgebungs-Subsysteme“ stellen die Ausführungsumgebungen für Anwendungsprogramme bereit. Es existieren mehrere Umgebungs-Subsysteme mit deren Hilfe die Eigenschaften fremder Betriebssysteme (zumindest teilweise) emuliert werden können. Das „POSIX-Subsystem“ ist zum Beispiel ein Umgebungssystem, das unter Windows NT die Ausführung von bestehenden POSIX-Anwendungen ermöglichen soll. Das „POSIX-Subsystem“ ist also für Kompatibilitätszwecke in Windows NT enthalten. (Die Kompatibilität bezieht sich dabei auf die Quelltexte der Anwendungen.) Das „Win32-Subsystem“ ist hingegen das *eigene* Umgebungssubsystem von Windows NT. Die „Systemdienste“ bilden die obere Schicht der Executive-Dienste und enthalten Schnittstellen zwischen dem Kernel-Modus und den „Subsystemen“ im Benutzermodus.
Windows NT repräsentiert interne Ressourcen abstrakt als Objekte. Dabei handelt es sich um Softwarekomponenten, die einen Datentyp besitzen und aus Attributen sowie einer Vielzahl von Operationen bestehen. So gibt es etwa

- Verzeichnisobjekte
- Dateiobjekte
- Port-Objekte
- Prozess- und Thread-Objekte.

Für die einheitliche Verwaltung der Objekte ist der Objekt-Manager verantwortlich. Er achtet auf die Einhaltung diverser Regeln, die die Benennung und Sicherheitsbehandlung von Objekten betreffen. Um von einem Programm (Prozess) auf ein Objekt zugreifen zu können, ist der Erhalt eines sog. „Objekt-Handles“ (eine Zugriffsnummer auf das Objekt) erforderlich. Die Erzeugung derartiger Handles ist ebenfalls eine Funktion des Objekt-Managers. Windows NT arbeitet mit einem einheitlichen Sicherheitssystem für den Schutz von Objekten.

5.2 Sicherheitskonzepte

Mit Windows NT lassen sich bezogen auf die Sicherheit verschiedene Abstufungen umsetzen, dabei reicht das Spektrum von „keine Sicherheit“ bis zu den Kriterien des „C2-Standards“ des US-amerikanischen *National Computer Security Center* (NCSC), bezogen auf Einzelplatzsysteme.
Der C2-Standard erfordert u.a. die Einhaltung folgender Kriterien ([Dap97], [MSD98]):

- Jeder Benutzer muss sich dem System gegenüber beim Anmelden (Login) mit einem eindeutigen Benutzernamen und einem Passwort identifizieren.
- Eine Überwachung aller sicherheitsrelevanten Aktionen ist möglich. Zugriff auf die Überwachungsdaten muss auf autorisiertes Personal beschränkt sein.

- Der Besitzer einer Ressource hat über diese die vollständige Zugriffskontrolle. Er kann anderen Benutzern und Benutzergruppen individuelle Berechtigungen gewähren oder entziehen.
- Auf gelöschte Objekte (z.B. Dateien) kann nicht noch einmal zugegriffen werden.
- Administratoren haben keine unbeschränkten Dateizugriffsrechte. Sie können bei Bedarf den Besitz einer Datei übernehmen, was der ursprüngliche Besitzer aber erkennen kann.

Jeder Benutzer (Mitarbeiter), der mit Windows NT arbeiten soll, benötigt ein Benutzerkonto (engl. User Account). Dieses Konto beinhaltet u.a. eine Benutzerkennung (den Namen für das Konto), ein Passwort und eine Sicherheits-identifikation (SID). Name und Passwort müssen vom Benutzer im Rahmen der Anmeldeprozedur angegeben werden. Die SID wird ausschließlich für interne Zwecke von Windows NT genutzt, für den Benutzer ist sie transparent. Auf weitere Details zu Benutzerkonten wird im Abschnitt 5.4 eingegangen.
Fast jedes Objekt[3] innerhalb von Windows NT kann Sicherheitsattribute erhalten. Diese Attribute werden durch einen Sicherheitsdeskriptor (engl. S*ecurity Descriptor*; Abk. SD) beschrieben, siehe Abbildung 20.

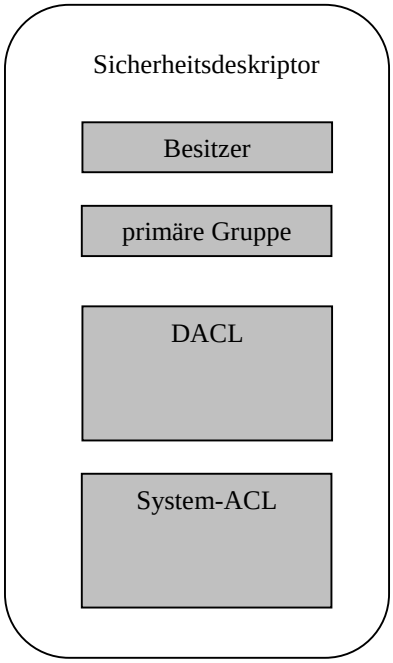

Abbildung 20: prinzipieller Aufbau eines Sicherheitsdeskriptors (nach [Meg98])

[3] Zumindest können alle „benannten“ Objekte, dazu gehören u.a. Dateien und Verzeichnisse, Sicherheitsattribute erhalten, für nähere Informationen sei auf [MSD98] verwiesen.

Ein solcher Deskriptor enthält vier Bestandteile:

- die SID des Besitzers dieses Objektes, Besitzer kann sowohl ein Benutzer als auch eine Gruppe sein.
- die SID der primären Gruppe, diese Information wird ausschließlich vom POSIX-Subsystem verwendet, sie hat für andere Umgebungssubsysteme keine Bedeutung.
- eine DACL
- eine System-ACL (SACL)

Eine Zugriffskontrolliste (engl. A*ccess Control List*; Abk. ACL) enthält Informationen über Zugriffsberechtigungen die erteilt, verweigert oder überwacht werden sollen. Die DACL (engl. Discretionary ACL) ist für die Festlegung von Zugriffsrechten für bestimmte Nutzer oder Gruppen auf Ressourcen relevant. Sie wird durch den Besitzer des Objektes kontrolliert. Die SACL (engl. System ACL) ist für die Überwachung von Objektzugriffen von Bedeutung. Für die Festlegung von SACLs ist der Systemadministrator verantwortlich. Die Nutzung von DACLs und SACLs ist sehr ähnlich. Ein Schwerpunkt der vorliegenden Arbeit ist die Verwaltung von Zugriffsrechten für Ressourcen, daher soll hier nur auf den Einsatz von DACLs eingegangen werden. Eine DACL kann Einträge in Form von ACEs aufnehmen. Jeder ACE (engl. Access Control Entry) enthält u.a.

- eine Angabe über einen Benutzer oder eine Gruppe, für welche(n) dieser ACE gilt, diese Angabe wird auch als „Trustee“ bezeichnet
- Zugriffsberechtigungen, die für den bzw. die Betreffenden gewährt oder verweigert werden.

Am Anfang einer DACL (im sog. „ACL-Header“) sind Verwaltungsinformationen zur Größe der ACL, zur Anzahl der enthaltenen ACEs und zur Revision der ACL gespeichert. Hinter dem ACL-Header befindet sich die Liste der ACEs, siehe Abbildung 21.

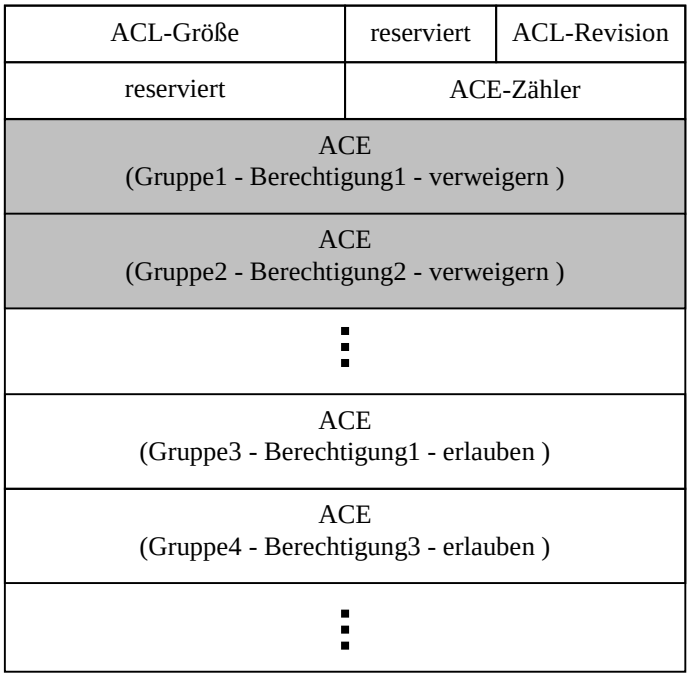

Abbildung 21: prinzipieller Aufbau einer DACL mit Beispiel-ACEs (nach [Zen97], [Meg98])

Bei der Verwendung von DACLs sind zwei Spezialfälle zu beachten:

- Falls ein SD keine DACL enthält, gibt es keine Zugriffsbeschränkungen für dieses Objekt, d.h. jeder Benutzer hat uneingeschränkten Zugriff darauf.
- Wenn der SD eine DACL besitzt, diese jedoch keine Einträge aufweist, so ist dadurch jeder Zugriff auf das Objekt explizit untersagt.

Am Anfang der ACE-Liste werden (unter Windows NT) all jene ACEs eingetragen, durch die Zugriffsberechtigungen explizit verweigert werden sollen. Beim Zugriff auf ein Objekt wird eine DACL immer so abgearbeitet, dass diese ACEs zuerst ausgewertet werden. Sie haben Priorität gegenüber ACEs die Zugriffsberechtigungen gewähren. Sobald bei der sukzessiven Auswertung der DACL durch einen ACE eine erforderliche Zugriffsberechtigung verweigert wird, wird der Zugriff auf das Objekt verweigert. Gegebenenfalls noch vorhandene ACEs werden dann nicht mehr untersucht, selbst wenn durch sie der Zugriff auf das Objekt mit der gewünschten Berechtigung erlaubt worden wäre.
Ein ACE besteht aus einem „ACE-Header“ (mit Angaben über Größe, Typ und Flags), einer Zugriffsmaske und der SID des Trustees, siehe Abbildung 22.

ACE-Größe	ACE-Flags	ACE-Typ
Zugriffsmaske		
SID (Trustee)		

Abbildung 22: Aufbau eines ACEs [Zen97]

Die ACE-Flags umfassen Verwaltungsinformationen, die sich auf die Vererbung von Berechtigungen beziehen und damit Objekttypen betreffen, die ihrerseits weitere Objekte enthalten können. In der Zugriffsmaske sind die Zugriffsberechtigungen für den Trustee enthalten. Es handelt sich um einen 32-Bit-Wert, der abhängig vom jeweiligen Objekttyp interpretiert wird. Der ACE-Typ legt fest, ob der ACE ein „Erlaubnis-ACE“ ist, durch den die Zugriffsberechtigungen genehmigt werden oder ob ein „Verbots-ACE“ vorliegt und die Zugriffsberechtigungen verwehrt werden.
Ein Überblick zu den Komponenten des Sicherheitssystems innerhalb der Architektur von Windows NT ist in Abbildung 23 dargestellt.

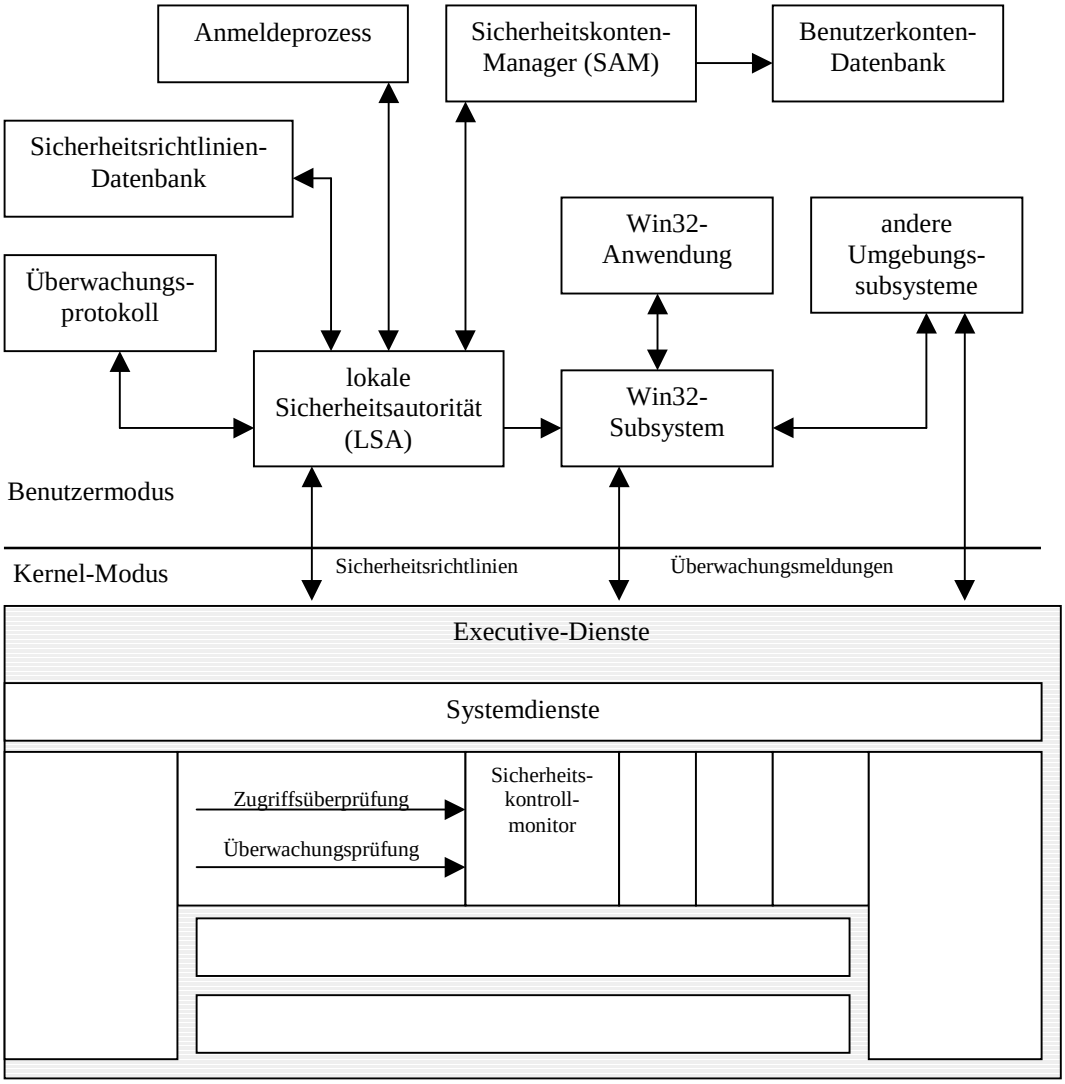

Abbildung 23: Sicherheitskomponenten von Windows NT (nach [Zen97], [MSD98])

Bei sicherheitsrelevanten Vorgängen werden entweder alle oder nur einzelne dieser Komponenten durchlaufen. Die „lokale Sicherheitsautorität“ (engl. Local Security Authority; Abk. LSA) ist der zentrale Bestandteil des Sicherheitssubsystems von Windows NT. Sie berücksichtigt die für den jeweiligen NT-Rechner konfigurierten Sicherheitsrichtlinien, die u.a. festlegen

- ob Datei- und Objektzugriffe der Benutzer überwacht werden sollen und
- wie lang Passworte der Benutzer mindestens sein müssen.

Falls z.B. die Überwachung von Objektzugriffen stattfinden soll, nimmt die LSA die vom Sicherheitskontrollmonitor generierten Meldungen entgegen und speichert diese im Überwachungsprotokoll. Im Folgenden sollen das Zusammenspiel und die Funktion der einzelnen Teile anhand des Anmeldevorgangs (siehe Abbildung 24) verdeutlicht werden.

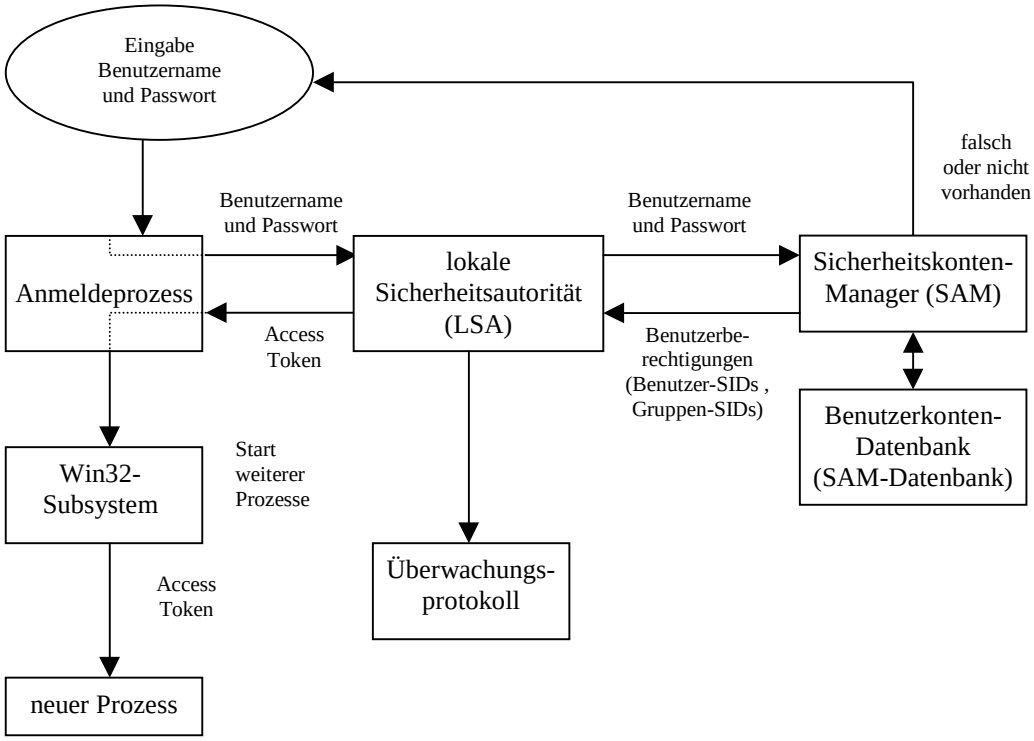

Abbildung 24: Abläufe beim Anmeldevorgang (nach [Zen97])

Der Anmeldeprozess nimmt die Benutzereingaben zu Benutzername und Passwort entgegen. Er leitet diese an die lokale Sicherheitsautorität weiter. Sie fragt nun beim Sicherheitskonten-Manager (engl. Security Accounts Manager, Abk. SAM) nach, ob dem System ein Benutzerkonto mit dem entsprechenden Namen und Passwort bekannt ist. Der SAM unterhält eine Datenbank, die in der Literatur teilweise als Verzeichnisdatenbank, Sicherheitsdatenbank, Benutzerkontendatenbank oder SAM-Datenbank bezeichnet wird. Wenn in dieser Datenbank kein entsprechender Eintrag gefunden wird, gilt der Anmeldeversuch als gescheitert. Der Benutzer erfährt, dass die Anmeldung nicht erfolgreich verlief. Ob der Benutzername oder das Passwort fehlerhaft waren, bleibt dem Benutzer verborgen. Er erhält die Möglichkeit, seine Eingaben zu korrigieren. Falls allerdings in der SAM-Datenbank ein passendes Konto existiert, liefert der SAM die Sicherheits-ID (SID) des Benutzers und die SIDs aller Gruppen, in denen der Nutzer Mitglied ist, an die lokale Sicherheitsautorität. Die LSA generiert nun eine Zugriffsmarke (engl. Access Token) und ordnet diese dem Benutzer zu. Diese Zugriffsmarke gilt, wie eine Art Fahr- oder Eintrittskarte, bis der Benutzer seine Arbeit beendet und sich beim System abmeldet. Sie enthält die ermittelten Sicherheits-IDs und entspricht damit den Berechtigungen des Benutzers. Die Gesamtheit der Berechtigungen des Benutzers wird auch als Sicherheitskontext bezeichnet. Jeder Prozess, der im Auftrag des Benutzers ausgeführt wird, erhält eine Kopie des Access Tokens und läuft somit

im Sicherheitskontext des Benutzers. Die Anmeldeprozedur ist damit beendet und die grafische Arbeitsumgebung wird für den Benutzer gestartet. Bei jedem Objektzugriff werden die im Access Token enthaltenen Informationen vom Sicherheitskontrollmonitor (engl. Security Reference Monitor) mit den Informationen im Sicherheitsdeskriptor des Objektes verglichen, um – jeweils entsprechend der ermittelten Berechtigung des Nutzers – den Zugriff zu gestatten oder zurückzuweisen.

5.3 Windows NT-Netzwerke

Um ein Windows NT-Netzwerk zu strukturieren, gibt es zwei grundsätzliche Konzepte. Zum einen können NT-Rechner logisch in einer sog. „Arbeitsgruppe“ zusammengefasst werden. Die andere Variante besteht im Anlegen einer „NT-Domäne“. Ein Rechner kann entweder Mitglied in genau einer NT-Domäne oder in genau einer Arbeitsgruppe sein, aber nie beides zugleich. Beide Konzepte ermöglichen den Mitarbeitern die Nutzung gemeinsamer Ressourcen (Drucker, Verzeichnisse). Allerdings bestehen in Bezug auf die Verwaltung der Computer gravierende Unterschiede.

5.3.1 Arbeitsgruppe

Microsofts Umsetzung des „Peer-to-Peer-Netzwerkprinzips“ (dt. etwa: „Gleicher-unter-Gleichen-Prinzip“) besteht in der Möglichkeit, mehrere Computer innerhalb einer Arbeitsgruppe zu organisieren. Eine solche Arbeitsgruppe kann sowohl Rechner mit der Workstation- als auch der Servervariante von Windows NT aufnehmen. Entsprechend dem Peer-To-Peer-Prinzip sind alle Mitgliedsrechner logisch gleichberechtigt. Jeder Rechner kann dabei Ressourcen für die Nutzung innerhalb der Arbeitsgruppe zur Verfügung stellen. Dies wird auch als Freigeben der Ressourcen bezeichnet. Jeder Rechner wird eigenständig verwaltet, d.h. vorhandene Einstellungen auf anderen Rechnern können nicht einfach einbezogen werden, sondern sind für jede einzelne Maschine neu zu konfigurieren. Um z.B. mehreren Nutzern das Arbeiten an einem Computer zu ermöglichen, muss für jeden Mitarbeiter ein Benutzerkonto auf der betreffenden Maschine eingerichtet und gepflegt werden. Soll das Passwort eines Nutzers geändert werden, so sind die damit verbundenen Verwaltungstätigkeiten auf jeder Maschine durchzuführen. Der Verwaltungsaufwand für die Rechner der Arbeitsgruppe steigt somit durch jeden zusätzlichen Computer beträchtlich. In gleichem Maße nimmt die Übersichtlichkeit ab, da es keine zentrale Verwaltungsmöglichkeit gibt. Arbeitsgruppen sind damit nur für geringe Mitarbeiterzahlen akzeptabel und werden im Folgenden nicht weiter betrachtet.

5.3.2 Domäne

Der Begriff *Domäne* hat mehrere Bedeutungen und wird in vielen Bereichen der Informatik zur Bezeichnung für sehr verschiedene Dinge verwendet. Im Zusammenhang mit Windows NT 4.0 steht er synonym für *NT-Domäne* und hat insbesondere nichts mit dem Namen- und Adressen-System innerhalb des Internets (dem DNS: Domain Name System) zu tun. Eine NT-Domäne entspricht einer Gruppe von Rechnern, die gemeinsam verwaltet werden können. Dazu übernimmt ein Computer eine herausragende Stellung als „Primärer Domänen-Controller“ (engl. Primary Domain Controller, Abk. PDC). Logisch gesehen übernimmt dieser Computer die Funktion eines Servers für die Domäne, die anderen Rechner der Dömane sind – gemäß dem Client-Server-Netzwerkprinzip – in der Rolle der Clients. Während auf den Client-Rechnern entweder die Workstation- oder die Server-Version von Windows NT installiert sein kann, ist für einen PDC das Betriebssystem „Windows NT Server“ erforderlich. Nur bei der Installation eines PDCs kann eine NT-Domäne angelegt werden. Er bietet die Möglichkeit einer zentralen Verwaltung von Benutzerkonten, die innerhalb der gesamten NT-Domäne gelten. Mitarbeiter haben so die Möglichkeit, sich von jedem anderen Mitgliedsrechner an der Domäne anzumelden. Der PDC ist als einziger Rechner der Domäne in der Lage, Benutzerkonten in so genannten „globalen Gruppen“ zusammenzufassen, die – wie der Name bereits andeutet – ebenfalls innerhalb der gesamten Domäne verwendet werden können. Außer einem PDC können in der Domäne weitere Rechner unter Windows NT-Server als sog. „Backup-Domänen-Controller“ (engl. Backup Domain Controller, Abk. BDC) oder als „Mitglieds-Server“ (engl. Member Server) eingerichtet werden. Ein BDC bekommt vom PDC durch Replikationsmechanismen eine Kopie von dessen SAM-Datenbank. Da ein BDC keinerlei Schreibrechte auf diese Kopie erhält, können auf ihm keine Benutzerkonten für die Domäne angelegt oder verändert werden. Allerdings kann auch er die Authentifikation (Echtheitsbestätigung) der Benutzer beim Anmeldevorgang übernehmen und dadurch den PDC entlasten. Bei Ausfall des PDCs – z.B. wegen Wartungsarbeiten – ist so der Arbeitsbetrieb der Mitarbeiter sichergestellt. Bei längerem Ausfall des PDCs kann ein BDC manuell durch den Systemverwalter zum PDC heraufgestuft werden. Mitglieds-Server haben bezogen auf die Verwaltung der Domäne keinen speziellen Status, sie gehören zu den Client-Rechnern der NT-Domäne. Jedoch dienen die Mitglieds-Server meist als Plattform für Applikationen, die mehreren Benutzern zugänglich gemacht werden sollen, wie beispielsweise Datenbanken.

5.3.3 Domänen-Modelle

Zwischen zwei NT-Domänen A und B kann eine sog. „Vertrauensstellung“ (engl.

„Trust Relationship“) eingerichtet werden. Dabei vertraut die Domäne A der Domäne B bzw. den Benutzern aus der Domäne B. Benutzer, die sich an der Domäne B angemeldet haben, können auf Ressourcen der Domäne A zugreifen. Die Domäne A wird allgemein „vertrauende Domäne“ (engl. „Trusting Domain“) genannt. Die Domäne B wird als „vertraute Domäne“ (engl. „Trusted Domain“) bezeichnet. Grafisch wird diese Vertrauensstellung durch einen Pfeil, der von A nach B gerichtet ist, veranschaulicht, siehe Abbildung 25.

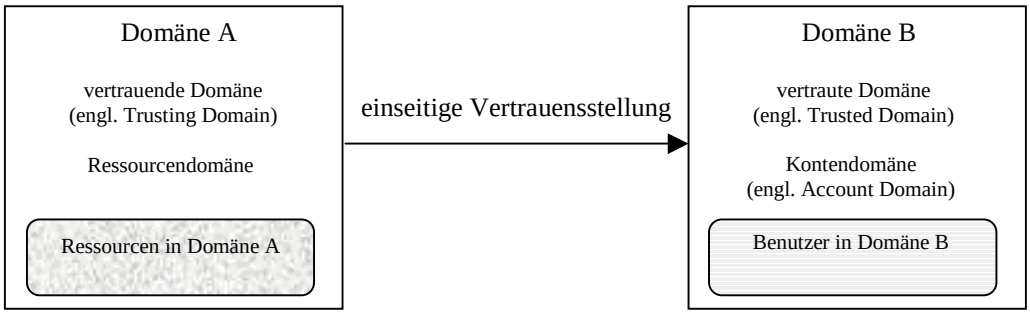

Abbildung 25: einseitige Vertrauensstellung zwischen den Domänen A und B

Aus der Perspektive der Domäne B stehen durch diese Vertrauensstellung die Ressourcen der Domäne A zur Verfügung. Die vertrauende Domäne A wird daher auch Ressourcendomäne genannt. Aus Sicht der Domäne A sind die Benutzerkonten und die Konten für globale Gruppen der Domäne B nutzbar. Die vertraute Domäne B wird daher auch als Kontendomäne bezeichnet. Wenn nur die Domäne A der Domäne B vertraut, wird dafür die Bezeichnung „einseitige Vertrauensstellung“ verwendet. Falls zusätzlich die Domäne B der Domäne A vertraut, wird dies „beidseitige“ oder „gegenseitige Vertrauensstellung“ genannt. Mit Hilfe von Vertrauensstellungen können einzelne Domänen zu komplexeren Verwaltungsstrukturen kombiniert werden, dadurch kann eine Anpassung der Netzwerkstrukturen an die Organisationsstruktur des Unternehmens erreicht werden. Zu beachten ist, dass Vertrauensstellungen nicht transitiv sind. Wenn eine Domäne A einer Domäne B vertraut und B einer Domäne C vertraut, so existiert dadurch keine Vertrauensstellung zwischen A und C.
Microsoft hat vier Domänen-Modelle als Musterkonfigurationen entwickelt [Tie98]:

- Einzeldomäne (engl. Single Domain)
- Hauptdomäne (engl. Master Domain)
- Mehrfachhauptdomäne (engl. Multiple Master Domain)
- Vollständige Vertrauensstellungen (engl. Complete Trust)

Jedes der vier Modelle genügt den folgenden zwei Kriterien:

- Für jeden Mitarbeiter muss nur ein Benutzerkonto eingerichtet werden.

- Jeder Mitarbeiter muss sich nur einmal an einer Kontendomäne anmelden, um entsprechend der für ihn erteilten Berechtigungen auf jegliche Ressourcen innerhalb des betreffenden Domänen-Modells zugreifen zu können.

Von diesen Modellen kann eines als Grundlage für die Strukturierung eines konkreten NT-Netzwerkes gewählt werden. Dabei spielen u.a. Überlegungen zu Performance- und Sicherheitsaspekten auf Grund

- der Anzahl der Benutzer und Computer,
- der Standort- und Abteilungsstruktur des Unternehmens sowie
- firmenpolitischer Entscheidungen

eine Rolle. Falls erforderlich sind – durch Veränderung bzw. Ergänzung der vorgeschlagenen Vertrauensstellungen – weitere Anpassungen des betreffenden Modells an die Bedürfnisse des Unternehmens durchführbar. Da aus der Wahl des Domänen-Modells Konsequenzen für die Verwaltung der Benutzer und Ressourcen resultieren, soll im Folgenden auf die Charakteristik der vier Modelle eingegangen werden.

5.3.3.1 Einzeldomäne

Dies ist ein einfaches Modell. Es existieren keine Vertrauensstellungen. Alle Benutzer- und Gruppenkonten, alle Ressourcen sind innerhalb einer Domäne definiert und werden zentral verwaltet. Allerdings besitzt dieses Modell einige Merkmale, die sich in Bezug auf die Sicherheit der IT-Umgebung negativ auswirken können:

- Eine physische Trennung von Ressourcen, z.B. eine abteilungsweise Aufgliederung, ist nicht möglich.
- Ähnlich verhält es sich mit den Benutzerkonten, auch hier gibt es keine Möglichkeit einer generellen Abgrenzung auf Abteilungsebene.
- Es ist nicht möglich sensible Ressourcen separat (von den Benutzern) zu halten, da alle Ressourcen und Benutzer in der gleichen Domäne definiert sind.
- Es ist nicht möglich „Subadministratoren“ zu definieren, die nur für die Verwaltung einiger bestimmter Gruppen zuständig sind. Dies ist eine generelle Schwäche des Domänen-Konzeptes von Windows NT 4.0, die auch in den anderen Domänen-Modellen zum Tragen kommt.

5.3.3.2 Hauptdomäne

Bei diesem Modell sind alle Benutzerkonten in einer Domäne – der Hauptdomäne – definiert. Dadurch ist eine zentrale Verwaltung der Benutzer gewährleistet. Die Ressourcen werden in separaten Domänen, den Ressourcendomänen angelegt. Diese können getrennt voneinander verwaltet werden. Es besteht jeweils eine einseitige Vertrauensstellungen zwischen einer Ressourcendomäne und der Hauptdomäne, siehe Abbildung 26.

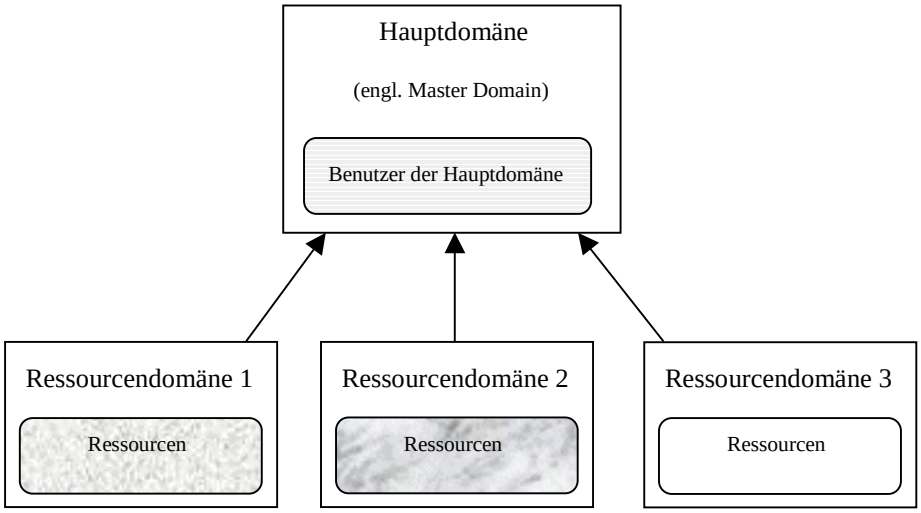

Abbildung 26: schematische Darstellung einer Hauptdomäne (nach [Tie98])

Im Vergleich zur Einzeldomäne ergeben sich für das Domänen-Modell „Hauptdomäne“ folgende Änderungen:

- Die physische Trennung von Ressourcen wird durch die Abgrenzung der einzelnen Ressourcendomänen ermöglicht. Allerdings erfordert jede Ressourcendomäne die Einrichtung eines eigenen PDCs, also zusätzlichen Aufwand für Verwaltung sowie Hard- und Software. In der Praxis wird dadurch die Flexibilität bei der Aufgliederung separater Ressourcenbereiche, z.B. anhand der Abteilungsstruktur, eingeschränkt.
- Benutzer und Ressourcen sind in getrennten Bereichen definiert.

5.3.3.3 Mehrfachhauptdomäne

Statt einer Hauptdomäne gibt es in diesem Modell mehrere Hauptdomänen, in denen die Benutzer(konten) definiert sind. Dies wirkt sich positiv auf die Skalierbarkeit dieses Domänen-Modells – bezogen auf die mögliche Anzahl der Benutzer und Computer – sowie das Leistungsverhalten der betroffenen Computersysteme und Netzwerke aus. Die beim Anmeldevorgang der Benutzer entstehenden Netz- und Systembelastungen sind durch die Hauptdomänen aufgeteilt. Zwischen den Hauptdomänen werden jeweils gegenseitige Vertrauensstellungen eingerichtet. Die Resourcendomänen vertrauen den Hauptdomänen, siehe Abbildung 27.

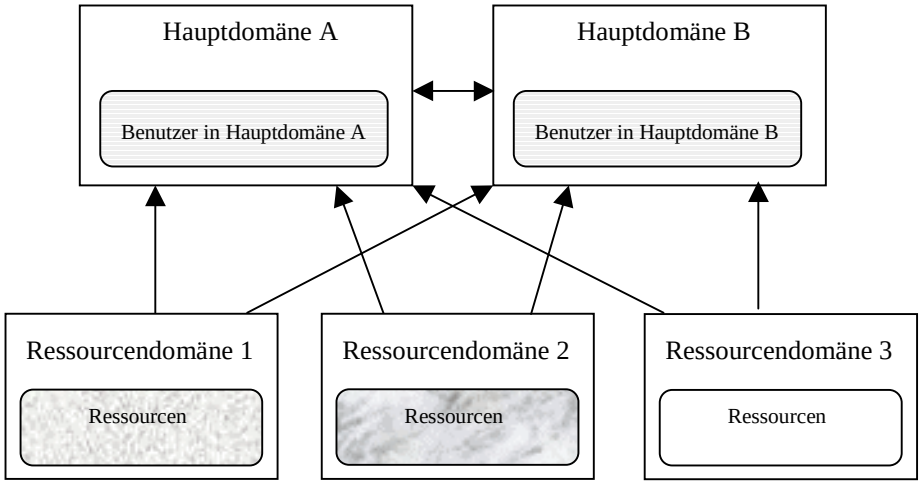

Abbildung 27: Mehrfachhauptdomäne (schematische Darstellung)

Das Domänen-Modell „Mehrfachhauptdomäne“ ist beispielsweise für umfangreiche Firmennetze, bei denen mehrere Firmenstandorte separat verwaltet werden sollen, besonders geeignet [Zen97]. Pro Standort wird eine Hauptdomäne eingerichtet, in der die Benutzerkonten für die (an diesem Standort ansässigen) Mitarbeiter angelegt werden. Gegenüber dem Modell mit einer Hauptdomäne besitzt dieses Modell einige veränderte Merkmale:

- Benutzerkonten sind zwischen den Hauptdomänen aufgeteilt.
- Es ist sowohl eine zentrale als auch eine dezentrale Benutzerverwaltung realisierbar.
- Es besteht ein erhöhter Verwaltungsaufwand aufgrund der zusätzlichen Vertrauensstellungen.
- Das Vorhandensein mehrerer Hauptdomänen hat negative Nebeneffekte für die Benutzerverwaltung, beispielsweise müssen globale Gruppen mehrfach angelegt werden. (Auf diese Problematik wird im Abschnitt 5.4 noch näher eingegangen.)

5.3.3.4 Vollständige Vertrauensstellungen

Wie der Name dieses Domänen-Modells bereits andeutet, vertraut hierbei jede Domäne jeder anderen Domäne. Zwischen den Domänen sind also jeweils beidseitige Vertrauensstellungen eingerichtet, siehe Abbildung 28.

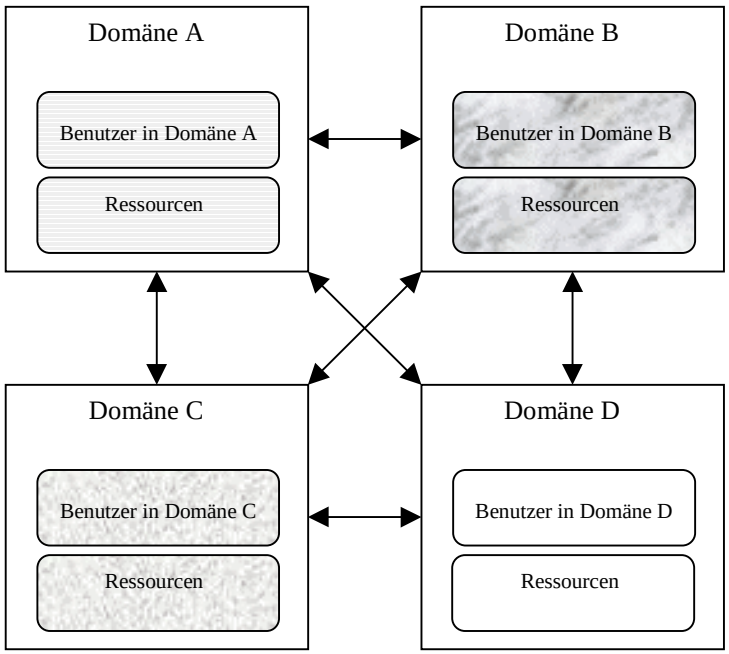

Abbildung 28: Vollständige Vertrauensstellungen (schematische Darstellung)

Unter Aspekten der Leistungsfähigkeit ist dieses Domänen-Modell sehr gut skalierbar. Unter Verwaltungsaspekten ist das Modell jedoch sehr problematisch, dieser Umstand verschärft sich mit zunehmender Anzahl von Domänen. Es besitzt u.a. folgende Eigenschaften:

- Jede Domäne enthält eigene Benutzer und Ressourcen. Dies ermöglicht eine abteilungsweise Aufgliederung und Verwaltung.
- Die maximierte Anzahl der Vertrauensstellungen verursacht einen erhöhten Verwaltungsaufwand.
- Eine zentrale Administration der Sicherheit ist sehr zweifelhaft. (Das betrifft insbesondere die Kontrolle, ob firmenweite Richtlinien und Empfehlungen eingehalten werden.)

5.4 Benutzer und Gruppen unter Windows NT

Beim Erstellen eines Benutzerkontos werden hauptsächlich Angaben zur Identifikation, zur Gruppenzugehörigkeit und zu Rechten des Benutzers gemacht. Die betreffenden Informationen werden in der SAM-Datenbank gespeichert. Dazu gehören u.a.

- der Benutzername (Anmeldename): Dies ist gewissermaßen die einzige obligatorische Angabe beim Anlegen des Benutzerkontos, die anderen Angaben sind entweder optional oder können durch Standardwerte belegt werden. Die maximale Länge des Anmeldenamens beträgt 20 Zeichen.

- der vollständige Name
- eine Beschreibung
- das Kennwort (Passwort): Es ist maximal 14 Zeichen lang.

Außerdem können folgende Optionen für das Benutzerkonto gewählt werden:

- Benutzer muss Kennwort bei der nächsten Anmeldung ändern
- Benutzer kann (darf) das Passwort nicht ändern
- Passwort läuft nie ab
- Konto deaktiviert

Ein Benutzerkonto kann mit obiger Option wahlweise deaktiviert werden. Dadurch wird das Benutzerkonto für den Benutzer stillgelegt, er kann sich damit nicht anmelden. Die Einstellungen des Benutzerkontos bleiben jedoch (in der SAM-Datenbank) erhalten, es kann bei Bedarf wieder aktiviert werden. Die Deaktivierungsmöglichkeit kann auch zum Einrichten von Musterkonten, die als eine Art Schablone für neu anzulegende Konten dienen [Meg98], verwendet werden.

Für bereits bestehende Benutzerkonten existiert zusätzlich die Option „Konto gesperrt“. Diese Option kann nicht direkt an- sondern nur abgewählt werden, um ein gesperrtes Konto zu entsperren. Die Sperrung eines Benutzerkontos kann beispielsweise dadurch verursacht werden, dass die (vom Administrator vorgegebene) zulässige Anzahl fehlerhafter Anmeldeversuche überschritten wurde.

Weitere Konfigurationseinstellungen eines Benutzerkontos umfassen Angaben für:

- den Pfad zum Basisverzeichnis (Home Directory)
- den Pfad zum Benutzerprofil
- den Pfad zum Anmeldeskript
- Anmeldezeiten
- Arbeitsstationen, von denen sich der Benutzer anmelden darf
- Ablaufdatum
- Kontotyp

Das Basisverzeichnis nimmt die benutzereigenen Dateien auf. Über Anmeldeskript und Benutzerprofil erfolgt die Konfiguration der grafischen Arbeitsumgebung für den Benutzer. Mit der Option „Anmeldezeiten“ kann detailliert festgelegt werden, wann der Benutzer sich an der Domäne anmelden bzw. Ressourcen der Domäne nutzen darf. Für das Benutzerkonto kann ein Ablaufdatum vorgegeben werden.

Beim Kontotyp wird angegeben, ob es sich um ein „globales Konto“ oder ein „lokales Konto“ handelt. Typischerweise werden Benutzerkonten in einer Domäne als „globales Konto“ angelegt. Ein „lokales Konto“ ist dagegen ein Spezialfall. Der Benutzer kann sich über das lokale Konto nicht (direkt) anmelden. Es wird verwendet, wenn zwischen zwei Domänen A und B keine Vertrauensstellung von A nach B besteht. Ein Benutzer, dessen normales „globales Konto“ in Domäne B definiert ist, kann auf Ressourcen in Domäne A zugreifen, wenn für ihn in Domäne A ein lokales Benutzerkonto eingerichtet wurde.

Intern wird von Windows NT für jedes Benutzerkonto eine SID generiert. Die SID ist einzigartig (für alle Zeit und Lokation), d.h. wird ein Benutzerkonto gelöscht und anschließend unter Verwendung identischer Angaben wieder erzeugt, so wird trotzdem eine andere SID zugeordnet. Das Umbenennen des Benutzerkontos ist möglich, die SID bleibt dabei erhalten. Windows NT verwendet die SIDs zur Verwaltung von Benutzern sowie deren Rechten und Berechtigungen. Windows NT unterscheidet zwischen Rechten und Berechtigungen (obwohl diese Begriffe im Allgemeinen eher synonym verwendet werden, die Literatur zu Windows NT macht dabei keine Ausnahme).
Rechte bestimmen die Aktionen, die ein Benutzer unter Windows NT ausführen darf. Sie werden daher auch Benutzerrechte genannt und haben Gültigkeit für das System, auf dem sie erteilt werden. Werden sie auf einem Domänen-Controller erteilt, so gelten sie für alle Domänen-Controller innerhalb der Domäne [MSD98]. Ansonsten gelten sie nur auf dem betreffenden NT-Rechner.
Berechtigungen beziehen sich auf bestimmte Ressourcen und werden auch als Objektberechtigungen oder Zugriffsberechtigungen bezeichnet. Die Benutzerrechte haben Priorität gegenüber den Objektberechtigungen und können diese gegebenenfalls überstimmen. In Windows NT 4.0 sind die Rechte fest vom System vorgegeben, sie können sowohl Benutzern als auch Gruppen erteilt oder versagt werden. Eine Ergänzung, durch vom Administrator individuell erstellte Rechte, ist nicht möglich. Eine Auswahl der existierenden Rechte liefert Tabelle 4.

Benutzerrecht	Bemerkung
Lokale Anmeldung	Der Benutzer darf sich lokal am NT-Rechner anmelden.
Zugriff auf diesen Computer vom Netz	
Herunterfahren des Systems	
Übernehmen des Besitzes an Dateien und Objekten	Der Benutzer erhält hierbei die Kontrolle über das betreffende Objekt.
Sichern von Dateien und Verzeichnissen	Erlaubt es dem Benutzer, eine Datensicherung der Dateien und Verzeichnisse vorzunehmen. Der Benutzer kann also sämtliche Dateien lesen, selbst wenn ihm die Objektberechtigungen dafür fehlen bzw. explizit entzogen wurden.
Wiederherstellen von Dateien und Verzeichnissen	Gestattet es, zuvor gesicherte Dateien und Verzeichnisse wiederherzustellen. Dieses Recht wirkt ebenfalls unabhängig von existierenden Zugriffsberechtigungen für die Dateien und Verzeichnisse.
Als Teil des Betriebssystems handeln	
Ändern der Systemzeit	Erlaubt dem Benutzer, die interne Uhr des NT-Rechners einzustellen

Tabelle 4: Beispiele für Benutzerrechte

Benutzerrechte und Zugriffsberechtigungen werden meist nicht für jedes Benutzerkonto einzeln vergeben, sondern einer Gruppe erteilt. Damit gelten sie für alle Mitglieder der Gruppe. Windows NT enthält bereits nach der Installation einige vordefinierte Benutzer- und Gruppenkonten, denen schon Benutzerrechte zugewiesen sind. Zugriffsberechtigungen für Ressourcen (insbesondere für Dateien und Verzeichnisse) müssen hingegen erst noch konfiguriert werden, da sie sich im Wesentlichen auf Verzeichnisse von Anwendungsprogrammen und zugehörigen Daten (jenseits des Betriebssystems) beziehen. Typischerweise werden in diesem Zusammenhang neue Gruppen eingerichtet.
Windows NT bietet drei Arten von Gruppen [Tie98]:

- globale Gruppen
- lokale Gruppen
- besondere/implizite Gruppen

„Globale Gruppen“ können nur auf einem PDC angelegt werden. Dazu wird ein Name für die globale Gruppe vergeben, dessen maximale Länge auf 20 Zeichen beschränkt ist. Der Name der Gruppe kann später nicht mehr geändert werden.
Intern generiert Windows NT für jedes Gruppenkonto – ebenso wie bei Benutzerkonten – eine eindeutige SID.
Optional kann für das Konto einer globalen Gruppe eine Beschreibung angegeben werden. Eine globale Gruppe kann ausschließlich Benutzerkonten der Domäne enthalten, in welcher sie angelegt wird. Sie kann also keine Benutzer aus verschiedenen Domänen vereinen, auch vorhandene Vertrauensstellungen zwischen den Domänen ändern daran nichts. (In Domänen-Modellen, bei denen es mehrere Kontendomänen gibt, müssen daher globale Gruppen gegebenenfalls mehrfach – d.h. in jeder Kontendomäne – angelegt werden.)
Globale Gruppen sind nicht in der Lage, andere Gruppen aufzunehmen. Globale Gruppen sind innerhalb einer Domäne von allen Rechnern aus sichtbar (d.h. ihnen können Benutzerrechte und Zugriffsberechtigungen erteilt werden), sie werden daher auch als „Domänen-Gruppen“ bezeichnet [Zen97]. Darüber hinaus erstreckt sich ihre Sichtbarkeit auf alle Rechner, die zu Ressourcendomänen gehören.
Der Geltungsbereich von „lokalen Gruppen“ hingegen, umfasst ausschließlich den jeweiligen NT-Rechner. Neben dem PDC, dessen SAM-Datenbank zur Speicherung der Benutzer- und Gruppenkonten innerhalb der Domäne dient, verfügen auch die Client-Rechner der Domäne (ebenso wie NT-Rechner, die nicht in einem Netzwerk betrieben werden) über eine lokale Benutzerverwaltung, deren Geltungsbereich auf den jeweiligen NT-Rechner beschränkt ist. Die betreffenden Informationen werden in einer lokalen SAM-Datenbank verwaltet und haben daher für andere Rechner in der Domäne keine Bedeutung. Lokalen Gruppen können Benutzerrechte und Zugriffsberechtigungen auf Ressourcen des jeweiligen Computers erteilt werden.

Lokale Gruppen können keine lokalen Gruppen enthalten. Sie sind allerdings in der Lage, sowohl

- Benutzerkonten, die nur auf dem lokalen Rechner definiert sind, als auch
- Benutzerkonten und Konten globaler Gruppen aus der zugehörigen Domäne (der Domäne, welcher der lokale Rechner angehört) sowie
- Benutzerkonten und Konten globaler Gruppen aus vertrauten Domänen (Kontendomänen)

als Mitglieder aufzunehmen. Der Name einer lokalen Gruppe kann bis zu 256 Zeichen lang sein. Es ist nicht möglich, lokale Gruppen umzubenennen. Auch für Konten lokaler Gruppen kann eine Beschreibung eingetragen werden.
In Tabelle 5 sind einige der vordefinierten Konten von Windows NT als Beispiele für Benutzer, globale Gruppen und lokale Gruppen aufgeführt.

Name des Kontos	Typ des Kontos	Bemerkung
Administrator	Benutzer	Jede Domäne und jeder Client-Rechner in der Domäne besitzen ein eigenes „Administrator“-Konto. Dieses Konto sollte nur für Verwaltungsaufgaben genutzt werden. Es ist Mitglied in der jeweiligen lokalen Gruppe „Administratoren“, aus der es nicht entfernt werden kann. Aus Sicherheitsgründen kann es sinnvoll sein, dieses Konto umzubenennen.
Domänen-Admins	globale Gruppe	Das „Administrator“-Konto der Domäne ist Mitglied dieser Gruppe. Sie wird auf jedem Rechner der Domäne Mitglied in dessen lokaler Gruppe „Administratoren“.
Administratoren	lokale Gruppe	Die Mitglieder dieser Gruppe verfügen über alle zur Verwaltung der Domäne bzw. des lokalen Rechners nötigen Rechte.
Domänen-Benutzer	globale Gruppe	Diese Gruppe enthält das „Administrator-Konto“ der Domäne und jedes neu hinzugefügte Benutzerkonto der Domäne. Diese Gruppe wird auf jedem Rechner der Domäne Mitglied in dessen lokaler Gruppe „Benutzer“.
Benutzer	lokale Gruppe	Diese Gruppe enthält alle „gewöhnlichen“ Benutzer.
Gast	Benutzer	Dieses Konto ist standardmäßig deaktiviert. Es existiert, wie beim „Administrator“-Konto, in jeder Domäne und auf jedem Client-Rechner in der Domäne ein eigenes Benutzerkonto „Gast“.
Domänen-Gäste	globale Gruppe	Das „Gast“-Konto der Domäne ist Mitglied dieser Gruppe. Sie wird auf jedem Rechner der Domäne Mitglied in dessen lokaler Gruppe „Gäste“.
Gäste	lokale Gruppe	Das lokale „Gast“-Konto eines Client-Rechners ist standardmäßig Mitglied in dieser Gruppe. Sie verfügt nur über eingeschränkte Rechte bzw. Zugriffsberechtigungen.

Tabelle 5: Beispiele für vordefinierte Benutzer und Gruppenkonten bei Windows NT

Bei den „impliziten Gruppen“ können keine Mitglieder hinzugefügt oder entfernt werden. Implizite Gruppen werden auch als „besondere Gruppen“ bezeichnet. Sie entsprechen vorgefertigten Gruppierungen von Benutzern, die aus Sicht des Betriebssystems nach allgemeinen Kategorien angeordnet sind. Tabelle 6 gibt einen Überblick zu impliziten Gruppen.

implizite Gruppe	Bemerkung
Netzwerk	Benutzer, die über das Netzwerk auf den NT-Rechner zugreifen, sind automatisch Mitglied dieser Gruppe.
Interaktiv	Zu dieser Gruppe gehört der Benutzer, sobald er sich lokal am NT-Rechner angemeldet hat.
Jeder	Diese Gruppe enthält – wie der Name bereits andeutet – alle Benutzer. Sie kann als Vereinigung der Gruppenmitglieder von Interaktiv und Netzwerk aufgefasst werden.
Ersteller-Besitzer	Diese Gruppe steht stellvertretend für den bzw. die Besitzer eines Objektes.
System	Diese Gruppe steht synonym für das Betriebssystem.
Authentifizierte Benutzer	Diese Gruppe ist erst ab dem „Servicepack 3 zu Windows NT 4.0“ verfügbar [Dap97]. Sie ist als eine Alternative zur Gruppe „Jeder“ eingeführt worden. Im Gegensatz zu „Jeder“ enthält sie keine Benutzer, die anonym angemeldet sind bzw. anonym auf den NT-Rechner zugreifen [MSD98].[4]

Tabelle 6: Überblick zu den „impliziten Gruppen“ von Windows NT

Die impliziten Gruppen können bei der Vergabe von Zugriffsberechtigungen und – mit Ausnahme von „System“ sowie „Ersteller-Besitzer“ – auch Benutzerrechten verwendet werden.
Microsoft empfiehlt bei der Verwendung von Benutzer- und Gruppenkonten im Zusammenhang mit der Zuordnung von Benutzerrechten und Zugriffsberechtigungen folgende Vorgehensweise [MSD98]:

- Benutzer werden in globalen Gruppen zusammengefasst,
- Benutzerrechte und Zugriffsberechtigungen werden an lokale Gruppen vergeben,
- globale Gruppen werden zu Mitgliedern in den lokalen Gruppen gemacht.

Bei der Benennung von Benutzer- und Gruppenkonten ist zu beachten, dass ein Benutzername weder einem Gruppennamen noch dem Namen eines anderen Benutzers innerhalb der SAM-Datenbank (entweder der Domäne oder des lokalen

[4] Solche anonymen Verbindungen bzw. Zugriffe resultieren meist als Nebeneffekt in Windows NT-Netzen während der Nutzung (von Werkzeugen) des Betriebssystems. Angenommen, es liegt eine einseitige Vertrauensstellung zwischen zwei Domänen vor und ein lokaler Administrator in der Ressourcendomäne nutzt ein grafisches Dienstprogramm zur Festlegung von Zugriffsberechtigungen für lokale Ressourcen. Das Dienstprogramm stellt dann (automatisch) eine anonyme Verbindung zur Kontendomäne her, um die Namen globaler Benutzer- und Gruppenkonten zu ermitteln (und diese dann in Form einer Auswahlliste grafisch aufbereiten zu können) [MSD98].

Rechners) gleichen darf. An dieser Stelle sei darauf hingewiesen, dass in der SAM-Datenbank der Domäne für jeden Computer, der dieser Domäne angehört, ein Computerkonto eingerichtet wird. Bei diesen Konten handelt es sich um Benutzerkonten, die intern durch speziell dafür vorgesehene Bits (Flags) gekennzeichnet sind und von Windows NT intern verwendet werden. Die Namen dieser Computerkonten entstehen durch Anfügen eines Dollarzeichens ($) an den Computernamen. Analog dazu wird auch für jede Ressourcendomäne ein solches Benutzerkonto erstellt. Für die Benutzerkonten in der SAM-Datenbank einer NT-Domäne gibt es insgesamt folgende Typen [MSD98], siehe Tabelle 7:

Konten	interne (engl.) Bezeichnung dieses Kontotyps
(globale) Benutzerkonten	NORMAL_ACCOUNT
Benutzerkonten mit lokalem Kontotyp	TEMP_DUPLICATE_ACCOUNT
Computerkonten für Domänen-Controller	SERVER_TRUST_ACCOUNT
Computerkonten für Client-Rechner (Mitglieds-Server oder Rechner mit der Workstation-Version von Windows NT)	WORKSTATION_TRUST_ACCOUNT
Konten für Ressourcendomänen	INTERDOMAIN_TRUST_ACCOUNT

Tabelle 7: Überblick zu den verschiedenen Typen von Benutzerkonten in der SAM-Datenbank einer NT-Domäne

Die Computerkonten und die Konten für Ressourcendomänen sind intern Mitglied der Gruppe „Domänen-Benutzer“. Die Eigenschaften dieser Konten werden allerdings von Windows NT (d.h. von den enthaltenen Werkzeugen) nicht angezeigt.

5.5 Zugriffsberechtigungen für Dateien und Verzeichnisse in Windows NT-Netzwerken

Damit Ressourcen eines NT-Rechners, wie z.B. Verzeichnisse oder (angeschlossene) Drucker, für die gemeinsame Nutzung in einem Windows NT-Netzwerk zur Verfügung stehen (also netzwerkseitig auf sie zugegriffen werden kann), müssen diese Ressourcen freigegeben werden. Dazu ist für die betreffende Ressource eine „Freigabe“ (engl. „Share“) einzurichten. Eine Freigabe ist bei Windows NT gekennzeichnet durch:

- den lokalen Pfad bzw. den Gerätenamen der Ressource
- einen Freigabenamen (der Netzwerkname der freigegebenen Ressource):

Die Ressource wird unter diesem Namen im Netzwerk sichtbar. Die maximale Länge des Freigabenamens beträgt 80 Zeichen [Zen97]. Endet der Freigabename mit einem Dollarzeichen ($), so wird diese Freigabe von den bei Windows NT enthaltenen Werkzeugen (bis auf wenige Ausnahmen[5]) nicht angezeigt. Derartige Freigaben werden als „versteckte Freigaben“ bezeichnet. (Wer eine Freigabe nutzen möchte, muss ihren Namen kennen.)

- einen optionalen Kommentar (eine Beschreibung): Auch diese Angabe ist vom Netzwerk aus sichtbar.
- eine Angabe zur Anzahl der Nutzer, die gleichzeitig auf die Freigabe zugreifen dürfen: Hier lässt sich eine konkrete Anzahl vorgeben oder festlegen, dass eine unbegrenzte Anzahl zulässig ist.
- den Ressourcentyp
- Zugriffsberechtigungen (Freigabeberechtigungen): Die hierdurch konfigurierten Zugriffsberechtigungen gelten nur, wenn über die Freigabe auf die betreffende Ressource zugegriffen wird. Die Freigabeberechtigungen gelten nicht, wenn lokal auf die betreffende Ressource zugegriffen wird.

Die Zugriffsberechtigungen, die für eine Freigabe erteilt werden können, sind vom jeweiligen Ressourcentyp abhängig. Der Ressourcentyp ist durch die freizugebende Ressource implizit vorgegeben, es handelt sich um einen Wert, der nicht festgelegt, sondern lediglich angezeigt werden kann. Im Zusammenhang mit Freigaben werden folgende Ressourcentypen unterschieden:

- Verzeichnis
- Named Pipe
- Drucker
- DFÜ-Warteschlange
- Ressource unbekannter Art

Im Folgenden soll ausschließlich auf Freigaben vom Ressourcentyp „Verzeichnis“ näher eingegangen werden. Beim Zugriff auf Freigaben eines Rechners werden meist Pfade entsprechend der „Universal Naming Convention“ (UNC) angegeben. Pfadangaben, die dieser Namenskonvention entsprechen, werden auch als UNC-Namen bezeichnet. UNC-Namen werden nach folgendem Schema gebildet:

```
\\Rechnername\Freigabename
```

Durch die Freigabe eines Verzeichnisses kann – im Rahmen der angegebenen Freigabeberechtigungen – auf sämtliche Dateien und Unterverzeichnisse in diesem Verzeichnis zugegriffen werden. Der UNC-Name wird dann entsprechend ergänzt.

[5] Der Befehl „net share“ ist beispielsweise so eine Ausnahme.

Damit auf Dateien über das (Windows NT-)Netzwerk zugegriffen werden kann, muss ein übergeordnetes Verzeichnis freigegeben werden, denn für eine Datei selbst kann keine Freigabe eingerichtet werden [Zen97].
Wird z.B. auf einem Rechner namens „Olymp“ das Verzeichnis „C:\Daten\Texte“ (dies ist der lokale Pfad) unter dem Freigabenamen „Texte“ freigegeben, so lautet der UNC-Name für diese Freigabe:

```
\\Olymp\Texte
```

Existiert eine Datei „C:\Daten\Texte\Formulare\Formular_XYZ.txt“, so ist sie unter folgendem UNC-Namen erreichbar:

```
\\Olymp\Texte\Formulare\Formular_XYZ.txt
```

Für Ressourcen vom Typ „Verzeichnis“ können pro Ressource auch mehrere Freigaben angelegt werden. Dabei gelten die jeweiligen Freigabeberechtigungen nur im Zusammenhang mit der entsprechenden Freigabe. Bereits mit der Installation von Windows NT werden einige Freigaben eingerichtet, die nur administrativen Zwecken dienen und daher als „Standardfreigaben“ oder „administrative Freigaben“ bezeichnet werden. Für sie können keine Freigabeberechtigungen erteilt werden. Administratoren haben jedoch Zugriff auf diese Freigaben. Administrative Freigaben sind von Windows NT intern durch ein spezielles Bit gekennzeichnet [MSD98]. Außerdem sind sie als „versteckte Freigaben“ eingerichtet, u.a. existiert für jedes Laufwerk eines NT-Rechners eine administrative Freigabe. Durch diese wird das Wurzelverzeichnis des jeweiligen Laufwerkes freigegeben, der Freigabename besteht aus dem Laufwerksbuchstaben und einem angefügten Dollarzeichen. Für obigen Rechner „Olymp“ wird beispielsweise das Verzeichnis „C:\“ durch die Standardfreigabe mit dem UNC-Namen:

```
\\Olymp\C$
```

repräsentiert. Tabelle 8 gibt einen Überblick zu den Freigabeberechtigungen für Verzeichnisse.

Freigabe-berechtigung	engl. Bezeichnung	Bemerkung
Kein Zugriff	No Access	Auf das freigegebene Verzeichnis kann nicht zugegriffen werden.
Lesen	Read	Dies beinhaltet folgende Möglichkeiten: • Die Namen von Dateien und Unterverzeichnissen im freigegebenen Verzeichnis können angezeigt werden • Es kann in Unterverzeichnisse gewechselt werden. • Der Inhalt von Dateien kann angezeigt werden. • Dateien (Anwendungsprogramme) können ausgeführt werden.
Ändern	Change	Diese Freigabeberechtigung enthält neben den Berechtigungen bzw. Möglichkeiten, die durch „Lesen“ erteilt werden, Folgendes: • Anlegen (Hinzufügen) von Dateien und Unterverzeichnissen • Der Inhalt von Dateien kann geändert werden. • Dateien und Unterverzeichnisse können gelöscht werden.
Vollzugriff	Full Control	Diese Freigabeberechtigung enthält alle Berechtigungen bzw. Möglichkeiten, die durch „Ändern“ erteilt werden. Falls das Dateisystem „NTFS“ eingesetzt wird, gelten (für NTFS-Dateien und -Verzeichnisse) zusätzlich folgende Möglichkeiten: • Ändern von Berechtigungen • Besitz übernehmen

Tabelle 8: Übersicht zu Freigabeberechtigungen für Verzeichnisse ([Zen97], [Tie98])

Windows NT unterstützt u.a. folgende Dateisysteme[6]:

- FAT (File Allocation Table)
- NTFS (New Technology File System)

Freigabeberechtigungen können bei beiden Dateisysteme vergeben werden. Allerdings ist Windows NT nur beim Einsatz von NTFS in der Lage, darüber hinaus für Dateien und Verzeichnisse einen Zugriffsschutz anzubieten, der sowohl bei Zugriffen über das Netzwerk als auch bei lokalen Zugriffen wirksam ist. Denn nur bei der Verwendung von NTFS werden die dazu nötigen Sicherheitsdeskriptoren für Verzeichnisse und Dateien zur Verfügung gestellt. Die damit konfigurierbaren Zugriffsberechtigungen für Dateien und Verzeichnisse werden auch als „Datei- und Verzeichnisberechtigungen“ oder „NTFS-Berechtigungen“ bezeichnet. Beim netzwerkseitigen Zugriff auf freigegebene Verzeichnisse gelten die NTFS-

[6] Durch das Dateisystem wird insbesondere bestimmt, wie Dateien und Verzeichnisse auf einem Datenträger (Speichermedium) gespeichert werden.

Berechtigungen zusätzlich zu den Freigabeberechtigungen, so dass die jeweils restriktivere Zugriffsberechtigung gilt. Bei den NTFS-Berechtigungen wird zwischen „Standardberechtigungen“ und „spezifischen Zugriffsberechtigungen“ unterschieden. Für die spezifischen Zugriffsberechtigungen werden auch die Bezeichnungen „individuelle Zugriffsberechtigungen“ oder „Einzelberechtigungen“ verwendet. Bei den Standardberechtigungen handelt es sich jeweils um Kombinationen aus spezifischen Zugriffsberechtigungen. NTFS unterstützt folgende spezifische Zugriffsberechtigungen, siehe Tabelle 9:

spezifische Zugriffsberechtigung	engl. Bezeichnung	Abkürzung
Lesen	Read	R
Schreiben	Write	W
Ausführen	Execute	X
Löschen	Delete	D
Berechtigungen ändern	Change Permissions	P
Besitz übernehmen	Take Ownership	O

Tabelle 9: Übersicht zu spezifischen Zugriffsberechtigungen (Einzelberechtigungen) innerhalb des Dateisystems NTFS

Der Benutzer, der eine Datei oder ein Verzeichnis anlegt, wird im Sicherheitsdeskriptor als Besitzer eingetragen. Eine Sonderbehandlung gilt für Benutzer, die Mitglied in der lokalen Gruppe „Administratoren“ sind, denn anstelle dieser Benutzer wird immer die lokale Gruppe „Administratoren“ als Besitzer vermerkt. Grundsätzlich hat der Besitzer einer Datei bzw. eines Verzeichnisses immer die Möglichkeit, die Zugriffsberechtigungen für diese Datei bzw. für dieses Verzeichnis zu konfigurieren. Er kann durch Erteilen der Zugriffsberechtigung „Berechtigungen ändern“ anderen Benutzern oder Gruppen gestatten, ihrerseits die Zugriffsberechtigungen zu konfigurieren, d.h. die entsprechende DACL zu bearbeiten. Die Zugriffsberechtigung „Besitz übernehmen“ erlaubt es dem betreffenden Trustee, sich als neuer Besitzer im Sicherheitsdeskriptor des Datei- bzw. Verzeichnisobjektes einzutragen. Die lokale Gruppe Administratoren ist immer in der Lage, den Besitz von Dateien sowie Verzeichnissen zu übernehmen. Die Semantik der spezifischen Zugriffsberechtigungen für Dateien ist in Tabelle 10 dargestellt.

Einzelberechtigung (Abkürzung)	Bedeutung für Dateien
Lesen (R)	Der Inhalt, die Attribute, die Berechtigungen und der Besitzer der Datei können angezeigt werden.
Schreiben (W)	Der Inhalt sowie die Attribute der Datei können geändert werden. Die Berechtigungen und der Besitzer der Datei können angezeigt werden.
Ausführen (X)	Die Ausführung des betreffenden Programms wird gestattet. Die Attribute, die Berechtigungen und der Besitzer der Datei können angezeigt werden.
Löschen (D)	Die Datei kann gelöscht werden.
Berechtigungen ändern (P)	Erlaubt das Ändern der Zugriffsberechtigungen für die Datei.
Besitz übernehmen (O)	Ein Besitzwechsel kann durchgeführt werden.

Tabelle 10: Übersicht zur Bedeutung der spezifischen Zugriffsberechtigungen (Einzelberechtigung) für Dateien ([Zen97], [Tie98])

Für Dateien und Verzeichnisse können auch „Attribute“ („Dateiattribute“) vergeben werden. Diese existieren und gelten unabhängig von den NTFS-Berechtigungen. Die Attribute gelten immer für alle Benutzer und Gruppen und haben Vorrang gegenüber den Zugriffsberechtigungen [Tie98]. So bewirkt beispielsweise das Attribut „Schreibgeschützt“ für eine Datei, dass die Datei nicht verändert werden kann. Für weitere Informationen zu den Attributen sei auf den Anhang B verwiesen.
Tabelle 11 gibt Auskunft zur Bedeutung der spezifischen Zugriffsberechtigungen für Verzeichnisse.

Einzelberechtigung (Abkürzung)	Bedeutung für Verzeichnisse
Lesen (R)	Der Inhalt des Verzeichnisses (d. h. die Namen und Attribute von enthaltenen Dateien und Unterverzeichnissen) sowie die Berechtigungen und der Besitzer des Verzeichnisses können angezeigt werden.
Schreiben (W)	Diese Zugriffsberechtigung ermöglicht das Erstellen von Unterverzeichnissen und Dateien im betreffenden Verzeichnis. Die Attributen des Verzeichnisses können verändert werden. Der Besitzer und die Berechtigungen des Verzeichnisses können angezeigt werden.
Ausführen (X)	Hierdurch wird der Wechsel in das Verzeichnis ermöglicht. Der Besitzer und die Berechtigungen des Verzeichnisses können angezeigt werden. Bemerkung: Um ein Programm starten zu können, genügt es, wenn die Berechtigung „Ausführen“ für die Datei und die Berechtigung „Lesen“ für das (enthaltende) Verzeichnis erteilt wird.
Löschen (D)	Diese Berechtigung ist zum Löschen des Verzeichnisses erforderlich. Bemerkung: Zum Löschen eines Verzeichnisses werden auch die Berechtigungen „Löschen“ für die Dateien im Verzeichnis benötigt. Um einzelne Dateien löschen zu dürfen, ist es nicht notwendig, die Berechtigung „Löschen“ für das Verzeichnis zu haben, es genügt die Berechtigung „Löschen“ für die jeweiligen Dateien.
Berechtigungen ändern (P)	Erlaubt das Ändern der Zugriffsberechtigungen für das Verzeichnis.
Besitz übernehmen (O)	Ein Besitzwechsel kann durchgeführt werden.

Tabelle 11: Übersicht zur Bedeutung der spezifischen Zugriffsberechtigungen für Verzeichnisse ([Zen97], [Tie98])

Die Einrichtung von Zugriffsberechtigungen für Dateien und Verzeichnisse erfolgt immer, indem Standardberechtigungen erteilt werden. Dadurch werden vordefinierte Kombinationen von spezifischen Zugriffsberechtigungen vergeben. Bei Bedarf können die so zugewiesenen Kombinationen von Einzelberechtigungen aber (für den konkreten ACL-Eintrag) durch Hinzufügen oder Entziehen von Einzelberechtigungen angepasst werden. Eine derartige Kombination von spezifischen Zugriffsberechtigungen (jenseits der Standardberechtigungen) wird als „beschränkter Zugriff“ bezeichnet. Für Dateien existieren folgende Standardberechtigungen, siehe Tabelle 12:

Standardberechtigung	englische Bezeichnung	enthaltene Einzelberechtigungen
Kein Zugriff	No Access	Keine
Lesen	Read	RX
Ändern	Change	RWXD
Vollzugriff	Full Control	Alle, also RWXDPO

Tabelle 12: Übersicht zu Standardberechtigungen für Dateien

Für Verzeichnisberechtigungen wird unterschieden zwischen:

- Zugriffsberechtigungen, die für das Verzeichnis selbst sowie für neu angelegte Unterverzeichnisse gelten, dafür wird auch die Bezeichnung „Berechtigungen für den Verzeichniszugriff“ verwendet.
- Zugriffsberechtigungen, die für im Verzeichnis neu angelegte Dateien gelten, diese werden auch „Berechtigungen für den Dateizugriff“, „Datei-Default-Berechtigungen“ oder „Initial-Berechtigungen für Dateien“ genannt.

Dateien und Unterverzeichnisse, die in einem Verzeichnis neu angelegt werden, „erben“ also ihre Zugriffsberechtigungen von dem (sie enthaltenden) Verzeichnis. Dadurch ist es nicht notwendig, die NTFS-Berechtigungen explizit für jede einzelne Datei separat konfigurieren zu müssen. Beim Vergeben von Verzeichnisberechtigungen existiert eine Option, um die Verzeichnisberechtigungen von gegebenenfalls bereits existierenden Unterverzeichnissen zu ersetzen. Eine weitere Option gibt an, ob die Zugriffsberechtigungen von bereits enthaltenen Dateien (durch die „Initial-Berechtigungen für Dateien“) ersetzt werden sollen. Für Verzeichnisse gibt es folgende Standardberechtigungen, siehe Tabelle 13:

Standardberechtigung	englische Bezeichnung	enthaltene Einzelberechtigungen für den Verzeichniszugriff	enthaltene Einzelberechtigungen für den Dateizugriff
Kein Zugriff	No Access	Keine	Keine
Anzeigen	List	RX	Nicht angegeben
Lesen	Read	RX	RX
Hinzufügen	Add	WX	Nicht angegeben
Hinzufügen und Lesen	Add&Read	RWX	RX
Ändern	Change	RWXD	RWXD
Vollzugriff	Full Control	Alle	Alle, RWXDPO

Tabelle 13: Übersicht zu Standardberechtigungen für Verzeichnisse

Soll für Verzeichnisberechtigungen ein „beschränkter Zugriff“ eingerichtet werden,

können sowohl die „Berechtigungen für den Dateizugriff“ als auch die „Berechtigungen für den Verzeichniszugriff“ angepasst werden. Die Kombination von Einzelberechtigungen für den Dateizugriff wird dabei als „beschränkter Dateizugriff“, die für den Verzeichniszugriff als „beschränkter Verzeichniszugriff“ bezeichnet. Sowohl für „beschränkter Verzeichniszugriff“ als auch für „beschränkter Dateizugriff“ kann „Vollzugriff“ spezifiziert werden. Soll in einem Verzeichnis keine Vererbung der „Initial-Berechtigungen für Dateien“ an neu angelegte Dateien stattfinden, kann als „beschränkter Dateizugriff“ auch ein „Nicht angegeben“ zugeordnet werden. Wie in Tabelle 13 zu sehen ist, erfolgt ebenfalls keine derartige Vererbung, falls die Standardberechtigung „Anzeigen“ oder „Hinzufügen“ für das Verzeichnis vergeben wurde.

Mit der Standardberechtigung „Vollzugriff“ erhält der Trustee im Rahmen der „Berechtigungen für den Verzeichniszugriff“ an einer Stelle etwas mehr Befugnisse, als die enthaltenen Einzelberechtigungen gestatten würden. Er besitzt mit „Vollzugriff“ auch die Erlaubnis, Dateien und leere Unterverzeichnisse in einem Verzeichnis zu löschen, bei denen er nicht über die Einzelberechtigung „Löschen“ verfügt. Mit der Summe der enthaltenen Einzelberechtigungen, also (RWXDPO) als „beschränkter Verzeichniszugriff“, ist dies nicht möglich [Zen97].

Soll auf eine Datei oder ein Unterverzeichnis zugegriffen werden, werden auch für sämtliche Verzeichnisse, die im Pfad zu dieser Datei (bzw. zu diesem Unterverzeichnis) enthalten sind, die NTFS-Berechtigungen beachtet.

In Windows NT-Netzwerken lassen sich die Zugriffsmöglichkeiten von Benutzern auf (freigegebene) Verzeichnisse und die darin enthaltenen Dateien mit Hilfe der NTFS-Berechtigungen sehr detailliert festlegen. Falls allerdings auf einem NT-Rechner die Zugriffsberechtigungen für Verzeichnisse und Dateien nicht explizit konfiguriert werden, besitzt standardmäßig die Gruppe „Jeder“ die Berechtigung „Vollzugriff“ für weite Teile der vorhandenen Verzeichnisse und Dateien von Windows NT. Insbesondere erhält diese Gruppe auch für neu angelegte Verzeichnisse und Dateien die Standardberechtigung „Vollzugriff“. Wird beim Freigeben eines Verzeichnisses auf die Angabe der Freigabeberechtigungen verzichtet, so gilt ebenfalls standardmäßig für die Gruppe „Jeder“ die Freigabeberechtigung „Vollzugriff“.

Wenn auf einem NT-Rechner das Benutzerkonto „Gast“ aktiviert wurde und kein Kennwort besitzt, so werden netzwerkseitige Zugriffe auf Freigaben dieses Rechners gegebenenfalls automatisch über das Gast-Konto abgewickelt. Ist z.B. auf einem PDC das Gastkonto aktiv, so können auch Benutzer, die in dieser Domäne (bzw. in einer zugehörigen Kontendomäne) nicht über ein eigenes Benutzerkonto verfügen, auf Freigaben der Domäne zugreifen (z.B. mit den Berechtigungen von „Jeder“). Daher sollte das Gast-Konto stets deaktiviert bleiben.

6 Sicherheitsmanagement mit „TME 10 Security Management“

6.1 Konzept der rollenbasierten Zugriffskontrolle

„TME 10 Security Management“ (TSecMan) verspricht, eine zentralisierte Verwaltung von Zugriffsberechtigungen über unterschiedliche Betriebssysteme hinweg, darunter UNIX und Windows NT, zu ermöglichen [Tiv1]. Um dies zu erreichen, unterhält TSecMan ein eigenes rollenbasiertes Sicherheitsmodell (kurz: Rollenmodell), das auf die Sicherheitssysteme der verwalteten Betriebssysteme abgebildet wird. Im Folgenden soll dieses Modell skizziert werden, siehe Abbildung 29.

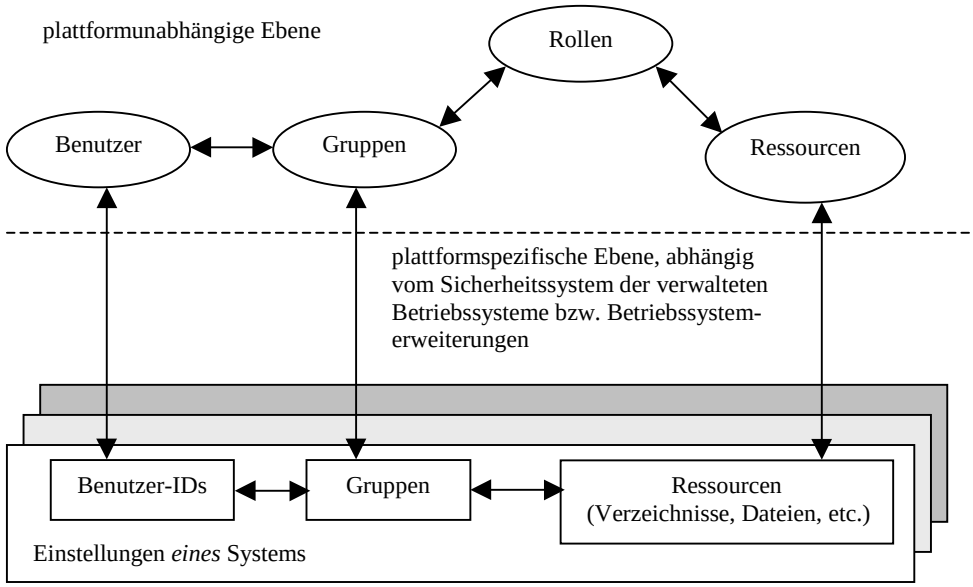

Abbildung 29: rollenbasiertes Sicherheitsmodell von „TME 10 Security Management“ (nach [Tiv9])

Die plattformunabhängige Ebene des Modells wird durch die Bestandteile Benutzer, Gruppen, Rollen und Ressourcen sowie die Beziehungen zwischen diesen charakterisiert. Diese Ebene entspricht der plattformübergreifenden Sicht auf das Unternehmen. Die Benutzer werden in Gruppen zusammengefasst. Die Mitglieder einer Gruppe sind jeweils für einen Teil der im Unternehmen zu leistenden Arbeit verantwortlich. Die Gruppenbildung kann beispielsweise anhand von Unternehmensbereichen, Abteilungen und Projektteams erfolgen [Tiv9], wobei jeder Benutzer auch in mehreren Gruppen Mitglied sein kann [Tiv1]. Die Gesamtheit aller Gruppen repräsentiert somit die Unternehmensstruktur. Um ihre Arbeitsaufgaben erledigen zu können, benötigen die Gruppen Zugriff auf Ressourcen. Als Ressourcen werden die

Objekte (in einem verteilten System) angesehen, für die ein Zugriffsschutz eingerichtet werden soll. Bei Ressourcen bleibt allerdings stets der Bezug zur jeweiligen Systemplattform erhalten, da jede Ressource durch den Plattformtyp (wie etwa „UNIX“ oder „Windows NT“), den Ressourcentyp und den Ressourcennamen identifiziert wird. Beispiele für Ressourcentypen sind insbesondere Dateien und Verzeichnisse, auf Ressourcentypen wird noch im Abschnitt 6.3.2 näher eingegangen. Die Zugriffsberechtigungen für Ressourcen werden nicht direkt an die Gruppen vergeben, sondern durch Zuordnung von Rollen zu Gruppen erteilt. Dabei dienen die Rollen gewissermaßen als Stellvertreter für Gruppen. Typischerweise werden pro Gruppe mehrere Rollen zugeordnet [Tiv1]. Die Rollen sind der zentrale Bestandteil des Sicherheitsmodells von TSecMan. Eine Rolle entspricht einer Zusammenstellung von Zugriffsberechtigungen für Ressourcen. Bei der Festlegung einer Rolle sollen genau die Zugriffsberechtigungen zusammengeführt werden, die für die Ausübung bestimmter Tätigkeiten erforderlich sind. Innerhalb einer Rolle werden somit Zugriffsberechtigungen entsprechend den Aufgabenbereichen der Mitarbeiter gesammelt. Beim Anlegen einer neuen Rolle, lässt sich angeben, dass diese von einer bereits existierenden Rolle abgeleitet werden soll. Die neue Rolle erbt damit die in der existierenden Rolle definierten Zugriffsberechtigungen. Für die neue Rolle können diese Berechtigungen dann eingeschränkt, erweitert oder (durch Hinzufügen von Zugriffsberechtigungen auf weitere Ressourcen) ergänzt, werden [Tiv9].
Bei der Abbildung der plattformunabhängigen Ebene des Rollenmodells auf die plattformspezifische Ebene nutzt TSecMan die Sicherheitsmechanismen der jeweils verwalteten Betriebssysteme. Im Fall von Windows NT erfolgt dies ohne die Verwendung von Betriebssystemerweiterungen[7]. Für Windows NT werden dabei folgende Zuordnungen vorgenommen, siehe Tabelle 14:

plattformunabhängige Sicht von TSecMan	verwendete Zuordnung unter Windows NT
Benutzer	Benutzer
Gruppen	globale Gruppen
Rollen	lokale Gruppen
Ressourcen	Ressourcen

Tabelle 14: plattformunabhängige Bestandteile des Rollenmodells von TSecMan und ihre Abbildung auf Windows NT

[7] Im Gegensatz zu UNIX, wo zusätzlich eine Komponente namens „Tivoli Access Control Facility“ (TACF) installiert wird, mit der eine Erweiterung der Zugriffskontrollmechanismen von UNIX umgesetzt wird und die u.a. eine eigene Datenbank zur Speicherung von Zugriffsinformationen unterhält [Tiv1].

TSecMan kann nur NT-Rechner verwalten, die in einer „Windows NT-Domäne“ (vgl. Abschnitt 5.3) organisiert sind [Tiv11].
Ein Vorzug des Rollenmodells liegt darin begründet, dass sich die Zugriffsberechtigungen, die benötigt werden um im Unternehmen eine bestimmte Arbeit zu erledigen, auch über längere Zeiträume hinweg kaum ändern. Wurden die Rollen definiert, entfallen so möglicherweise fehlerträchtige Vorgänge bei denen von einem Systemverwalter ad hoc Zugriffsberechtigungen an neu eingerichtete Gruppen oder gar Benutzer vergeben werden. Erhält beispielsweise ein Mitarbeiter aufgrund einer Beförderung innerhalb des Unternehmens eine andere Aufgabe, so genügt es, das Benutzerkonto aus den bisherigen Gruppen zu entfernen und zum Mitglied in einer anderen Gruppe zu machen. Auf diese Weise erhält der Mitarbeiter automatisch die für seine neue Tätigkeit erforderlichen Zugriffsberechtigungen. Durch das Vorhandensein von Rollen ist es möglich, die Administration der Zugriffsberechtigungen für Ressourcen einerseits von der Administration der Benutzerkonten andererseits so zu entkoppeln, dass die Verantwortung, die mit diesen Arbeitsschritten verbunden ist, auf mehrere Administratoren, die sich speziell einem der beiden Gebiete widmen, verteilt werden kann.

6.2 Zusammenarbeit mit anderen Managementapplikationen von Tivoli

TSecMan ist eine profilbasierte Managementapplikation, die auf dem TMF aufsetzt, d.h. TSecMan bedingt das Vorhandensein einer Installation des „Tivoli Management Framework“ (vgl. Kapitel 4). Die vorhandene TMF-Umgebung wird bei der Installation von TSecMan u.a. um

- Sicherheitsprofile (eine neue Profilart „Security Profile“)
- neue Kommandos für das CLI
- neue Privilegien, mit deren Zuweisung Administratoren innerhalb der TMR für die Nutzung von TSecMan autorisiert werden können

erweitert ([Tiv1], [Tiv11]).
Die grafische Bedienung von TSecMan erfolgt über den „TME 10 Desktop“. TSecMan macht allerdings keinen Gebrauch vom RIM, d.h TSecMan bietet keine Unterstützung für eine externe Datenbank.
TSecMan kann mit (einigen) anderen Managementapplikationen zusammenarbeiten, wobei TSecMan auch separat betrieben werden kann. Prinzipiell könnte TSecMan als einzige Managementapplikation innerhalb einer TMF-Installation genutzt werden. In realen Einsatzszenarios sind typischerweise außer TSecMan noch weitere Managementapplikationen vorhanden. (Auf Planungsaspekte für den Einsatz von TSecMan wird im Abschnitt 6.4 eingegangen.) TSecMan enthält, neben der Verwaltung von Zugriffsberechtigungen, u.a. auch die Möglichkeit, die

Sicherheitsprotokolle (security logs) der verwalteten Betriebssysteme auszuwerten und innerhalb von TSecMan zusammenzuführen. So können z.B. Zugriffe bzw. Zugriffsversuche auf Ressourcen zentral überwacht und protokolliert werden. In diesem Zusammenhang bietet TSecMan Kooperationsmöglichkeiten mit den Managementapplikationen [Tiv11]:

- „TME 10 Distributed Monitoring“: um beispielsweise die Größe der Protokolldateien auf den verwalteten Systemen zu überwachen und
- „TME 10 Enterprise Console“ (TEC): um Meldungen zu sicherheitskritischen Systemereignissen an die TEC weiterzuleiten, wo dann (nach vorher definierten Regeln) die Korrelation des Problems stattfinden kann.

Darüber hinaus ist insbesondere eine Integration zwischen TSecMan und „TME 10 User Administration“ möglich. „TME 10 User Administration“ (nachfolgend als „TUA“ abgekürzt) erweitert die TMF-Umgebung u.a. um Benutzerprofile (eine neue Profilart „User Profile“). Im Rahmen der Zusammenarbeit zwischen TSecMan und TUA können von TSecMan aus auch Informationen aus den Profilen von TUA genutzt werden.
Der folgende Abschnitt soll auf grundsätzliche Eigenschaften der Sicherheitsprofile eingehen und (einige) Details im Zusammenhang von Zugriffskontrolle und Benutzerverwaltung für – per TSecMan verwaltete – Windows NT-Netzwerke verdeutlichen.

6.3 Sicherheitsprofile

Sicherheitsprofile können Datensätze speichern, die für das Sicherheitsmanagement relevante Informationen repräsentieren. Sicherheitsprofile können bis zu vier unterschiedliche Datensatztypen (engl. „record types“) enthalten[8]. Dabei handelt es sich um folgende Datensatztypen:

- „Group“ (Gruppe)
- „Role“ (Rolle)
- „Resource“ (Ressource)
- „System Policy“ (Systemrichtlinie)

Innerhalb eines Sicherheitsprofils existieren die „Validation Policy“ und die „Default Policy“ pro Datensatztyp. Die Datensatztypen enthalten mehrere Attribute.

[8] Die Profilart „Security Profile“ ist – im Gegensatz zu anderen Profilarten – in der Lage, unterschiedliche Datensatztypen aufzunehmen [Tiv11].

Jeder Datensatztyp besitzt die Attribute:

- „Name“: Diese Angabe ist obligatorisch und dient zur Identifizierung des Datensatzes. Für Datensätze vom Typ „Resource“, muss die Kombination aus Plattformtyp, Ressourcentyp und Ressourcenname innerhalb des Sicherheitsprofils eindeutig sein. Für die anderen Datensatztypen gilt, dass der Name des Datensatzes (bezogenauf den Datensatztyp) innerhalb des Profilmanagers, der das betreffende Sicherheitsprofil enthält, eindeutig sein muss.
- „Description“: (eine optionale Beschreibung, von bis zu 255 Zeichen Länge)
- „RecordType“: Dieses Attribut ist durch den Datensatztyp implizit vorgegeben.

Einige Attribute beziehen sich nur auf einen Plattformtyp (z.B. auf UNIX), während andere Attribute nicht plattformspezifisch sind. Plattformspezifische Attribute gelten damit nur für die Profilendpunkte, auf denen das entsprechende Betriebssystem installiert ist. Daher werden plattformspezifische Attribute auch als endpunktspezifische Attribute bezeichnet. Manche Attribute dienen dazu, Beziehungen zwischen Datensätzen unterschiedlicher Datensatztypen herzustellen. Im Folgenden sollen exemplarisch für jeden Datensatztyp ein paar ausgewählte Attribute vorgestellt werden, eine umfassendere Darstellung ist in [Tiv11] zu finden.

6.3.1 Gruppendatensätze

Gruppendatensätze (Datensätze vom Typ „Group“) enthalten u. a. die Attribute:

- „Name“
- „NTName“
- „UXName“
- „TMEUserMembers“
- „NTUserMembers“
- „UXUserMembers“
- „Roles“

Die plattformspezifischen Attribute „NTName“ und „UXName“ besitzen per Voreinstellung den Wert des Attributes „Name“. Das Attribut NTName kann zur Anpassung des Gruppennamens für Windows NT genutzt werden, indem für dieses Attribut explizit ein anderer Wert als im Attribut „Name“ festgelegt wird. Analog dazu gilt „UXName“ für UNIX. Das Attribut „TMEUserMembers“ dient zur Integration von TSecMan mit TUA. Es kann eine Liste von Benutzernamen enthalten, die Mitglieder in der Gruppe werden. Benutzer, die in diese Liste eingetragen werden sollen, müssen in einem Benutzerprofil von TUA definiert worden sein. Beim Zusammenstellen der Liste können Benutzer aus verschiedenen Benutzerprofilen ausgewählt werden. Alternativ oder in Ergänzung

zu „TMEUserMembers“ kann das Attribut „NTUserMembers“ eine Liste mit Namen von Benutzern aufnehmen, die von einem NT-Rechner (Domänen-Controller) stammen (d.h. bereits existieren müssen) und zu Mitgliedern der Gruppe werden.
Mit TSecMan lassen sich keine neuen Benutzer(konten) definieren. TSecMan legt zwar für jede globale Gruppe einer NT-Domäne (die über einen Gruppendatensatz in TSecMan verwaltet wird) ein Benutzerkonto auf dem PDC der NT-Domäne an. Dieses Benutzerkonto wird allerdings von TSecMan nur für interne Zwecke genutzt und ist deaktiviert [Tiv11]. Der Name eines solchen Kontos ist aus dem Präfix „tme_“ und dem angefügten Gruppennamen zusammengesetzt.
TUA ist in der Lage, neue Benutzerkonten (auch für Windows NT) zu erstellen. Die Verwaltung der Gruppen erfolgt für Windows NT nicht mit TUA [Tiv12], sondern per TSecMan. Wird ein Sicherheitsprofil an einen PDC verteilt, legt TSecMan für neu erstellte Gruppendatensätze je eine globale Gruppe an.
Die Verwendung des Attributes „NTUserMembers“ ist insbesondere in jenen TMF-Umgebungen sinnvoll, in denen TUA nicht installiert ist oder nicht zur Benutzerverwaltung von Windows NT genutzt wird. Das Attribut „UXUserMembers“ ist ähnlich verwendbar wie „NTUserMembers“, bezieht sich aber auf UNIX-Systeme. Das Attribut „Roles“ enthält die Rollennamen, die dieser Gruppe zugeordnet wurden.

6.3.2 Ressourcendatensätze

Ressourcendatensätze (Datensätze vom Typ „Resource“) enthalten u. a. folgende Attribute:

- „Name“
- „EpType“
- „ResType“
- „DefAccess“
- „Roles“

Im Attribut „EpType“ wird der Endpunkttyp (Plattformtyp) der Ressource vermerkt. Als Endpunkttyp kann z.B. der Wert „NT“ für NT-Rechner oder „UX“ für UNIX-Rechner angegeben werden. Das Attribut „ResType“ enthält den Ressourcentyp. Die in TSecMan modellierbaren Ressourcentypen sind teilweise nur für einen Endpunkttyp verfügbar.
Im Attribut „Name“ von Ressourcendatensätzen kann nicht eine beliebig gewählte Bezeichnung verwendet werden. Es dient dazu, eine oder mehrere Ressourcen zu identifizieren und zwar sowohl im *Sicherheitsprofil* als auch auf dem *Profilendpunkt*. Daher ist das Format des Ressourcennamens auch vom Ressourcentyp abhängig. Innerhalb eines Sicherheitsprofils dürfen Ressourcendatensätze nur dann gleiche

Ressourcennamen besitzen, wenn die Ressourcen von unterschiedlichem Typ sind. TSecMan erstellt beim Verteilen von Sicherheitsprofilen (Distributing) keine neuen Ressourcen auf den Profilendpunkten [Tiv11]. Mit dem Attribut „DefAccess“ können Zugriffsberechtigungen für die Ressource vorgegeben werden. Die so definierten Zugriffsberechtigungen gelten auch dann (für Zugriffe von Benutzern), wenn (für diese Benutzer) keine Zugriffsberechtigungen aus Rollendefinitionen zutreffen. Beim Endpunkttyp NT wird DefAccess als Zugriffsberechtigung für die implizite Gruppe „Jeder“ abgebildet [Tiv10]. Zugriffsberechtigungen sind abhängig vom Ressourcentyp, daher ist die Verwendung des Attributes „DefAccess“ ebenfalls vom Ressourcentyp abhängig. Das Attribut „Roles“ enthält die Rollennamen, die Zugriffsberechtigungen für diese Ressource festlegen.
Für den Endpunkttyp NT bietet TSecMan u.a. folgende Ressourcentypen:

- DIRECTORY: Hierdurch kann ein Verzeichnis (nur NTFS) geschützt werden, wobei sich die Zugriffsberechtigungen dieser Ressourcendatensätze nur auf die „Berechtigungen für den Verzeichniszugriff“ des NTFS-Verzeichnisses beziehen (vgl. Abschnitt 5.5). Als Ressourcenname wird der vollständige Pfad zu dem Verzeichnis, inklusive Laufwerksbuchstabe, verwendet, also z.B. „C:\Daten\Texte“. (Es werden keine Platzhalterzeichen (Wildcards) unterstützt [Tiv11].)
- FILE: Mit diesem Ressourcentyp lassen sich Zugriffsberechtigungen für Dateien (nur NTFS) konfigurieren. Als Ressourcenname wird der vollständige Pfad zur Datei, inklusive Laufwerksbuchstabe, verwendet. Als Platzhalterzeichen darf ein „*“ am Ende des Ressourcennamens (nach einem Verzeichnis) stehen [Tiv11]. Wird im Ressourcennamen kein Platzhalterzeichen genutzt, lautet er z.B. „C:\Daten\Texte\Formulare\Formular_XYZ.txt“, beziehen sich Zugriffsberechtigungen für den Ressourcendatensatz auf die NTFS-Berechtigungen für die Datei. Wird hingegen das Platzhalterzeichen verwendet, z.B. im Ressourcennamen „C:\Daten\Texte\Formulare*“, so entsprechen die Zugriffsberechtigungen für den Datensatz den „Berechtigungen für den Dateizugriff“ des jeweiligen NTFS-Verzeichnises (im obigen Beispiel des Verzeichnisses „Formulare“).
- SHARE: Ein Ressourcendatensatz dieses Typs repräsentiert eine Verzeichnisfreigabe. Als Ressourcenname wird der Freigabename verwendet. Zugriffsberechtigungen für solche Ressourcendatensätze entsprechen den Freigabeberechtigungen.
- SYSTEM: Mit diesem Ressourcentyp werden die *Benutzerrechte* (vgl. Abschnitt 5.4) von Windows NT modelliert. Daher gibt es typischerweise nur einen Datensatz dieses Typs im jeweiligen Sicherheitsprofil. Falls mehrere solche Ressourcendatensätze an einen NT-Rechner verteilt werden, gilt nur der zuletzt

verteilte Ressourcendatensatz (da sie sich jeweils überschreiben). Der Ressourcenname kann bei diesem Typ beliebig gewählt werden.

TSecMan nutzt eigene (Bezeichnungen für) Zugriffsberechtigungen, wobei einige Zugriffsberechtigungen für mehr als einen Endpunkttyp und Ressourcentyp angewendet werden können. In Tabelle 15 sind die Zugriffsberechtigungen für Ressourcen(datensätze) dargestellt, die TSecMan – in Bezug auf den Endpunkttyp NT – für die Ressourcentypen DIRECTORY, FILE und SHARE verwendet.

Zugriffsberechtigung unter TSecMan	Abkürzung für Nutzung mit CLI	Äquivalente NTFS-Berechtigung (enthaltene Einzel-berechtigungen)	entsprechende Freigabe-berechtigung
Read	R	R	Lesen
Write	W	W	
Execute	X	X	
Delete	D	D	
Update	U	RWXD	Ändern
Full Control	F	Alle	Vollzugriff
Change Ownership	O	O	
Change Permissions	P	P	
No Access	N	Keine	Kein Zugriff

Tabelle 15: Übersicht (zu Bezeichnungen) der Zugriffsberechtigungen von TSecMan und ihre Bedeutung/Entsprechung unter Windows NT (nach [Tiv10], [Tiv11])

Auf Ressourcendatensätze des Typs SHARE ist nur eine der Zugriffsberechtigungen „Read“, „Update“, „Full Control“ oder „No Access“ anwendbar.

6.3.3 Rollendatensätze

Rollendatensätze (Datensätze vom Typ „Role“) enthalten insbesondere die Attribute:

- „Name“
- „TMEGroups“
- „NTGroups“
- „NTTMEResAccess“
- „NTResAccess“
- „Parent“

Das Attribut „Name“ enthält den Rollennamen. Das Attribut „TMEGroups“ entspricht einer Liste von Gruppennamen, denen diese Rolle zugeordnet wird. In diese Liste werden nur solche Gruppen aufgenommen, für die in TSecMan bereits Gruppendatensätze angelegt wurden. Alternativ oder in Ergänzung zu „TMEGroups“ kann mit dem Attribut „NTGroups“ eine Liste mit Namen von globalen Gruppen zusammengestellt werden, die ebenfalls dieser Rolle zugeordnet werden, aber bereits in einer NT-Domäne existieren müssen. Die globalen Gruppen dürfen aus verschiedenen NT-Domänen stammen. (Bei jedem Gruppennamen in dieser Liste kann der Name der betreffenden NT-Domäne mit angegeben werden.)
Die Attribute „NTTMEResAccess“ und „NTResAccess“ dienen zur Definition von Zugriffsberechtigungen für Ressourcen. Beide Attribute gelten für den Endpunkttyp NT. Beide Attribute definieren jeweils eine Liste, bei der jeder Eintrag aus einer Ressourcenbezeichnung und zugehörigen Zugriffsberechtigungen – für die Ressource(n) – besteht. Die Ressourcenbezeichnung umfasst den Ressourcentyp und den Ressourcennamen. Die Liste in „NTTMEResAccess“ kann nur solche Ressourcen enthalten, für die Ressourcendatensätze vorhanden sind. Bei den Ressourcenbezeichnungen in dieser Liste kann zusätzlich der Name des entspechenden Sicherheitsprofils spezifiziert werden. In die Liste von „NTResAccess“ lassen sich nur jene Ressourcen aufnehmen, die nicht als Ressourcendatensatz modelliert wurden.
Mit dem Attribut „Parent“ kann der Name eines bereits existierenden Rollen-datensatzes angegeben werden, dessen Einstellungen in die aktuelle Rolle übernommen werden. TSecMan legt – beim Verteilen eines Sicherheitsprofils an NT-Rechner – pro neu erstellter Rolle eine gleichnamige lokale Gruppe an und modifiziert die Zugriffsberechtigungen der jeweiligen Ressourcen.

6.3.4 Systemrichtliniendatensatz

Ein Systemrichtliniendatensatz (Datensatz vom Typ „System Policy“) enthält diverse Attribute, die den Anmeldevorgang der Benutzer (login) an die verwalteten Systeme betreffen. Typischerweise wird nur ein Systemrichtliniendatensatz definiert, der dann für alle Profilendpunkte gilt. Der Systemrichtliniendatensatz stellt u.a. folgende Attribute bereit:

- „Lockout“
- „NTLockout“
- „PwMinLen“
- „NTPwMinLen“

Das Attribut „Lockout“ legt Folgendes fest:

- ob Benutzerkonten gesperrt werden sollen, falls vom jeweiligen Benutzer

wiederholt ein fehlerhaftes Passwort angegeben wird
- die maximale Anzahl fehlerhafter Anmeldeversuche, die vor einer Sperrung stattfinden müssen
- welche Zeitspanne dem Benutzer eingeräumt wird, sich doch noch erfolgreich anzumelden.
- die Dauer der Sperre

Mit „NTLockout“ lassen sich diese Einstellungen für den Endpunkttyp NT anpassen. Die minimale Länge eines Passwortes kann mit dem Attribut „PwMinLen“ allgemein vorgegeben werden, während das Attribut „NTPwMinLen“ nur für Windows NT gilt.

6.3.5 Anlegen von Datensätzen in Sicherheitsprofilen

Die im Abschnitt 4.3 dargestellten Zusammenhänge für das Arbeiten mit Profilen gelten auch für TSecMan und die Sicherheitsprofile. An dieser Stelle soll daher hauptsächlich auf einige Details und Einschränkungen eingegangen werden, die das „Populating“ betreffen. Weil Sicherheitsprofile unterschiedliche Datensatztypen enthalten können, wurde das Populating entweder für alle Datensatztypen oder speziell für einen Datensatztyp vorgesehen. Das Populating eines Datensatztypes kann durch die explizite Angabe des Namens auf einen einzelnen Datensatz eingeschränkt werden. Die weiteren Ausführungen beziehen sich auf den Endpunkttyp NT, betreffen also das Abfragen von NT-Rechnern mittels „Populating“.
Gruppendatensätze müssen – selbstverständlich – von einem (oder mehreren) Domänen-Controller(n) abgefragt werden. Pro globaler Gruppe wird im Sicherheitsprofil ein Gruppendatensatz erstellt bzw. aktualisiert. Mitglieder der globalen Gruppe werden im plattformspezifischen Attribut (NTUserMembers) des jeweiligen Gruppendatensatzes eingetragen. Bei Ressourcendatensätzen ist das Populating jedoch mit Einschränkungen verbunden. Für den Endpunkttyp NT steht im Rahmen des Populating nur der Ressourcentyp SYSTEM zur Verfügung [Tiv11]. Dabei wird ein Datensatz namens „User Rights“ angelegt.
Auch für die Erfassung von Informationen zu lokalen Gruppen ist das Populating nur bedingt geeignet. TSecMan erfasst durch das Populating nur die von Windows NT vordefinierten lokalen Gruppen, d.h. nur lokale Gruppen die bei jeder Installation von Windows NT standardmäßig angelegt werden (z.B. die lokalen Gruppen „Benutzer“ und „Administratoren“). Für diese lokalen Gruppen wird je ein Rollendatensatz mit der Bezeichnung der jeweiligen lokalen Gruppe angelegt.
Ein weiteres Problem bei der Gewinnung von Informationen über bestehende Konfigurationen ist dadurch gegeben, dass vom Populating nur solche Informationen

erfasst werden, die zum rollenbasierten Sicherheitsmodell von TSecMan passen. Falls z.B. Benutzerkonten direkt in lokalen Gruppen enthalten sind (und nicht indirekt über eine globale Gruppe), wird dies nicht berücksichtigt.
Ein Systemrichtliniendatensatz kann per Populating erstellt bzw. aktualisiert werden [Tiv11].
Zusammenfassend kann festgestellt werden, dass TSecMan mittels Populating gewissermaßen eine Anfangsmenge von Datensätzen in einem Sicherheitsprofil anlegen kann. Die so gewonnenen Informationen, enthalten allerdings nur einen Bruchteil der sicherheitsrelevanten Aspekte und Einstellungen einer bereits bestehenden Windows NT-Domäne. Insbesondere fehlen sämtliche Informationen, die die Zugriffskontrolle für Verzeichnisse und Dateien betreffen sowie Informationen über solche lokale Gruppen, die individuell angelegt wurden. Für eine Vervollständigung des Sicherheitsprofils müssen die vorhandenen Datensätze editiert und durch neu anzulegende Datensätze ergänzt werden.
Um beispielsweise die Zugriffsberechtigungen für ein (im Windows NT-Netzwerk) freigegebenes Verzeichnis im Allgemeinen modellieren zu können, müssen einerseits die drei Ressourcentypen

- SHARE (Freigabeberechtigung),
- DIRECTORY (NTFS-Verzeichnisberechtigung) und
- FILE (NTFS-Dateizugriff)

genutzt werden. Andererseits sind nach Bedarf ein oder mehrere Rollendatensätze zu modifizieren (falls vom Populating angelegt) bzw. neu anzulegen. Dabei können die genannten Ressourcentypen als Ressourcendatensätze im Sicherheitsprofil erstellt werden. Diese Variante ermöglicht in jedem Ressourcendatensatz die Vorgabe von Zugriffsberechtigungen, die für die Gruppe „Jeder“ gelten sollen. Zugriffsberechtigungen für Rollen werden dann im Attribut „NTTMEResAccess“ spezifiziert. Alternativ dazu können die Ressourcen mit den entsprechenden Zugriffsberechtigungen im plattformspezifischen Attribut (NTResAccess) des jeweiligen Rollendatensatzes eingetragen werden.
In jedem Fall müssen die Ressourcennamen, also der Freigabename (für SHARE) und der Pfad des Verzeichnisses (jeweils bei DIRECTORY und FILE) eingegeben werden. Dies gilt sowohl für die Verwendung der GUI als auch für die Benutzung des CLI von TSecMan.
Ob beim Einsatz von TSecMan die Administration von Zugriffsberechtigungen ausschließlich über plattformspezifische Attribute der Rollendatensätze erfolgen soll, oder ob Gruppendatensätze und Ressourcendatensätze mit einbezogen oder ausschließlich verwendet werden, muss für jeden Anwendungsfall in Abhängigkeit von den konkreten Einsatzbedingungen und Einsatzzielen entschieden werden.

6.4 Generelle Betrachtungen zur Einführung von TSecMan (zum Sicherheitsmanagement einer IT-Umgebung)

Die Stärke des Rollenmodells und von TSecMan liegt in seiner Fähigkeit, Zugriffsberechtigungen über verschiedenen Plattformen hinweg in einer konsistenten Art und Weise administrieren zu können. Dabei wird durch das Rollenmodell eine strukturierte Herangehensweise an die Benutzerverwaltung und die Zugriffskontrolle forciert. Im Hinblick auf Windows NT-Netzwerke lässt sich mit TSecMan das Sicherheitsmanagement für mehrere NT-Domänen zentralisieren, ohne dass dafür Vertrauensstellungen zwischen diesen Domänen vorhanden sein müssen [Tiv10].
Der Hauptaufwand bei der Einführung bzw. Nutzung von TSecMan – unter Berücksichtigung einer bestehenden IT-Umgebung – betrifft die vorbereitenden Planungsaktivitäten im Vorfeld des eigentlichen Einsatzes. Ein sorgfältiger Entwurf des Rollenmodells bildet die Voraussetzung für ein – im Sinne des Sicherheitsmanagements – erfolgreiches Arbeiten mit TSecMan.
Wird TSecMan in eine bestehende EDV-Umgebung eingebracht, so kann im Idealfall auf ein bereits existierendes Rollenmodell zurückgegriffen werden. Im Rahmen der Einführung von TSecMan wird dieses Rollenmodell gegebenenfalls weiter verfeinert und schließlich in Sicherheitsprofilen modelliert. So kann es mit Hilfe der zentralisierten Kontroll- und Steuerungsmöglichkeiten von TSecMan auch auf andere Systeme ausgedehnt werden.
Für die Fälle, in denen das Rollenmodell nicht bereits explizit definiert bzw. formuliert wurde, sondern noch erarbeitet oder verfeinert werden muss, kommt es darauf an, sowohl

- die bestehenden System/Server-Konfigurationen
  als auch
- die unter dem Aspekt der EDV-Nutzung im Unternehmen bestehenden Aufgabengebiete, Arbeitsabläufe und -prozesse

zu erfassen und zu analysieren, um damit

- eine Grundlage für das zu erstellende Sicherheitsmodell zu erhalten,
- eventuell „implizit vorhandene“ Rollen zu erkennen und
- aktuelle Mängel und Schwachpunkte aufdecken zu können.

In jedem Fall ist es unabdingbar, den aktuellen Zustand der eingerichteten Zugriffsberechtigungen zu ermitteln und zu dokumentieren. Zumindest in Bezug auf Windows NT bietet TSecMan (mittels „Populating“) keine ausreichende Unterstützung zur Lösung dieser Aufgaben.
An dieser Stelle sei nochmals darauf verwiesen, dass das Vorhandensein (bzw. die Erarbeitung und die Pflege) einer „Sicherheitspolitik“ (vgl. Abschnitt 2.2.5) eine

unverzichtbare Basis für das gesamte Sicherheitsmanagement (und damit generell für den Einsatz eines Sicherheitsmanagement-Produktes) darstellt. Auch in [Tiv10] wird die Bedeutung der Sicherheitspolitik des jeweiligen Unternehmens für den Einsatz von TSecMan hervorgehoben. Dabei wird die Einführung einer auf TSecMan basierenden Lösung für das Sicherheitsmanagement der IT-Landschaft eines Unternehmens in vier Schritte bzw. Ablaufphasen eingeteilt. Abbildung 30 zeigt den schematischen Überblick zu diesen Ablaufphasen.

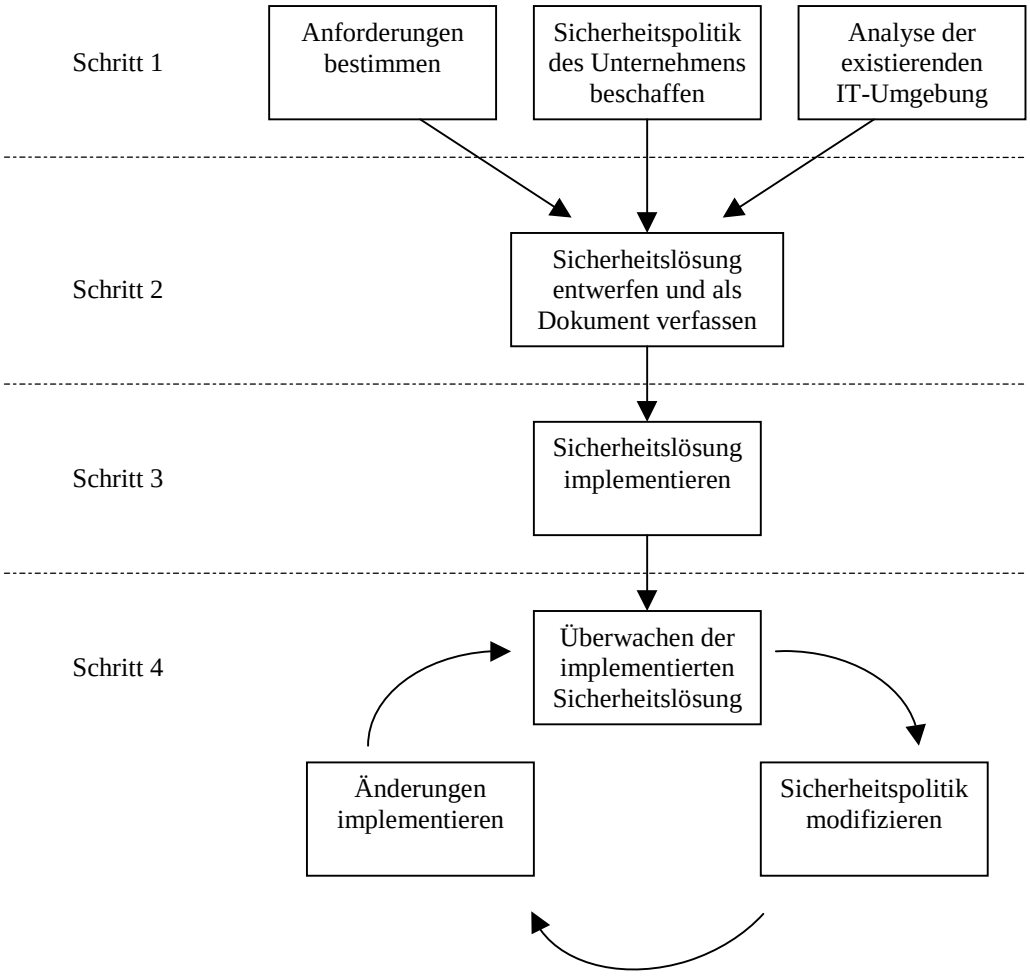

Abbildung 30: schrittweise Methode für Entwurf und Implementierung einer auf TSecMan basierenden Lösung für das Sicherheitsmanagement der IT-Umgebung eines Unternehmens (nach [Tiv10])

Im ersten Schritt müssen zunächst die Anforderungen und Erwartungen des Unternehmens an den Einsatz von TSecMan definiert werden. Sollen etwa bestimmte aufgetretene Sicherheitsmängel beseitigt werden oder geht es hauptsächlich darum, die bereits vorhandenen Sicherheitsmechanismen von einer zentralen

Stelle aus steuern zu können? Außerdem muss in Schritt 1 das bestehende Sicherheitsmanagement analysiert werden. Falls die Sicherheitspolitik des Unternehmens nicht als Dokument vorliegt, sollte ein solches Dokument angefertigt werden. Eine weitere Teilaufgabe von Schritt 1 besteht in der Analyse der vorhandenen IT-Umgebung und der tatsächlich implementierten Sicherheitsmaßnahmen. Dabei soll u.a. herausgefunden werden, ob die Sicherheitspolitik mit den vorhandenen Maßnahmen im Einklang steht und wo eine Integration zwischen TSecMan und anderen Produkten benötigt wird. Im Schritt 2 wird anhand der Ergebnisse aus Schritt 1 ein Entwurf für die benötigte Sicherheitslösung erstellt, dessen Formulierung sich an Konfigurationsoptionen von TSecMan orientiert. Basierend auf dem in Schritt 2 verfassten Dokument, erfolgt im Schritt 3 die Implementierung der entworfenen Sicherheitslösung. Der vierte Schritt steht für das Betreiben und Überwachen der implementierten Lösung. Dabei geht es um die Suche nach Sicherheitslöchern und Verbesserungsmöglichkeiten der Sicherheitslösung. Daraus resultieren Anpassungen der Sicherheitspolitik sowie die Umsetzung dieser Anpassungen. Schritt 4 verdeutlicht, dass durch die Einführung von TSecMan kein statischer Endzustand erreicht, sondern vielmehr ein dynamischer Prozess etabliert wird.
TSecMan ist also ein Werkzeug, das Administratoren dabei hilft, die konkrete Sicherheitspolitik für die IT-Umgebung des Unternehmens um- bzw. durchzusetzen.

7 Modellierungstool

7.1 Allgemein

In diesem Kapitel wird das „Modellierungstool“, eine Softwarelösung für die Ermittlung von (einigen) Informationen über konkrete Windows NT-Umgebungen, vorgestellt. Das Modellierungstool wurde als Bestandteil der vorliegenden Arbeit entworfen, implementiert und getestet. Es ist zur Nutzung durch Personen gedacht, die für das Sicherheitsmanagement in Windows NT-Netzwerken zuständig sind (also durch das Administrationspersonal der IT-Umgebung eines Unternehmens oder durch externe IT-Dienstleister). Das Modellierungstool soll die System-verantwortlichen bei der Bearbeitumg von Analyse- und Dokumentationsaufgaben unterstützen, die u.a. als ein Teilschritt bei der Erarbeitung eines Rollenmodells für bestehende Windows NT-Umgebungen anfallen. Das Modellierungstool ist in der Lage, diverse Informationen zum Ist-Zustand einer Windows-NT-Domäne in Bezug auf vorhandene

- Benutzer,
- lokale Gruppen,
- globale Gruppen und
- Freigaben

zu erfassen, grafisch darzustellen und zu speichern. Vom Modellierungstool wird also kein fertiges Rollenmodell für die Umsetzung in TSecMan entworfen. Die Bezeichnung „Modellierung“ bezieht sich hier darauf, dass die erfassten Konfigurationsinformationen (in einem plattformneutralen Format) in einer Datenbank gespeichert werden und danach unabhängig von Zugriffen auf die verwalteten NT-Rechner zu Verfügung stehen. Die Konfiguration der verwalteten NT-Rechner wird vom Modellierungstool nicht beeinflusst.
Das Modellierungstool wurde als Java-Programm implementiert. Es enthält im Wesentlichen drei Bestandteile:

- eine grafische Oberfläche (GUI),
- einen plattformspezifischen Programmteil (JNI-Teil) und
- einen Programmteil für die Verbindung zu einer Datenbank (JDBC-Teil).

Die grafische Oberfläche des Modellierungstools wurde unter Verwendung der „Swing-Klassen“ von Java umgesetzt. Der plattformspezifische Teil enthält Windows NT-spezifische Systemaufrufe, die entsprechend der „JNI-Spezifikation“ (JNI: „Java Native Interface“) in ein Java-Objekt gekapselt wurden. Der datenbank-bezogene Programmteil des Modellierungstools nutzt die JDBC-Technologie von Java (JDBC: „Java Database Connectivity“). Im JNI-Teil des Modellierungstools

werden die Informationen über die zu untersuchende Windows NT-Umgebung beschafft. Der JDBC-Teil dient dazu, diese Informationen zu speichern. Auf grundsätzliche Erläuterungen der Verwendung von Java und Einzelheiten zu Swing, JNI und JDBC soll an dieser Stelle verzichtet werden. Informationen zu diesen Themen sind in entsprechender Literatur, u.a. in [SUN1], [SUN2], zu finden.
Prinzipiell können die Bestandteile des Modellierungstools

- zu einer einzeln ausführbaren Java-Anwendung oder
- zu einer verteilten Java-Anwendung

kombiniert werden. Der plattformspezifische Teil muss auf einem NT-Rechner ausgeführt werden. Dabei verlässt sich dieser Teil auf die Privilegien und Berechtigungen (den Sicherheitskontext) des angemeldeten Benutzers.
Die Variante einer einzeln ausführbaren Anwendung wird im Folgenden auch als Szenario A bezeichnet und ist in Abbildung 31 dargestellt.

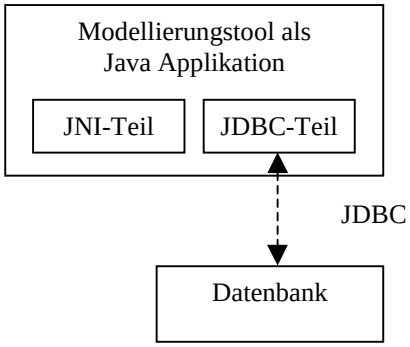

Abbildung 31: Modellierungstool als einzeln ausführbare Anwendung (schematische Darstellung von Szenario A)

Dieses Szenario wurde insbesondere bei der Entwicklung des JNI-Teils und des JDBC-Teils sowie den damit verbundenen Testläufen des Modellierungstools genutzt. Abbildung 32 zeigt das Schema von Szenario B, bei dem das Modellierungstool als verteilte Anwendung unter Nutzung von CORBA konzipiert wurde.

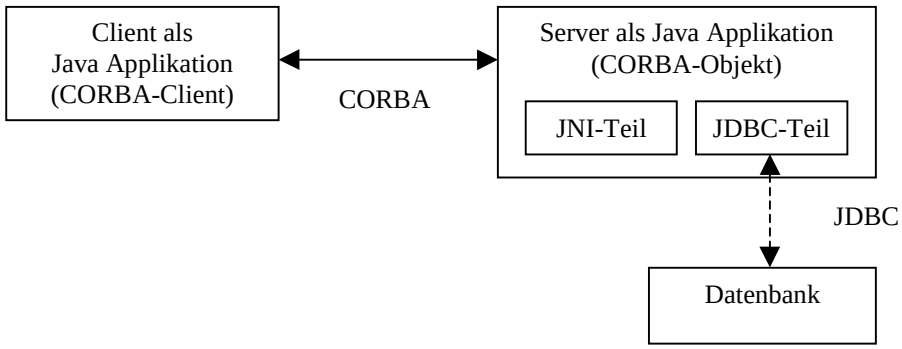

Abbildung 32: Modellierungstool als verteilte Anwendung mit Client als Java-Applikation (schematische Darstellung von Szenario B)

Szenario B wurde insbesondere zu Entwicklungs- und Testzwecken der auf CORBA basierenden Programmteile des Modellierungstools verwendet. (Beim Modellierungstool kommen synchrone Methodenaufrufe zum Einsatz, Client und Objektimplementierung nutzen statische Schnittstellen, vgl. Abschnitt 3.3.)
Abbildung 33 enthält eine schematische Darstellung von Szenario C, bei dem – im Unterschied zu Szenario B – der Client als Java-Applet ausgeführt ist.

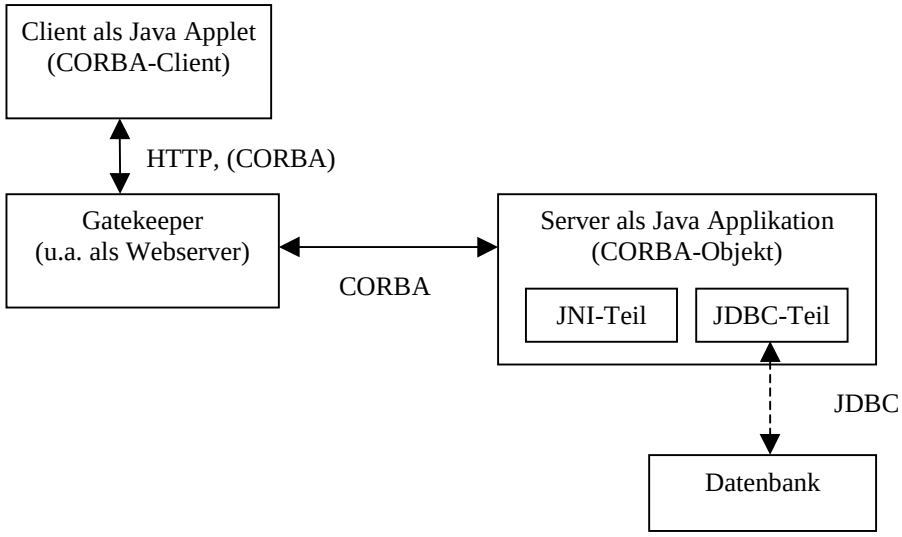

Abbildung 33: Modellierungstool als verteilte Anwendung mit Client als Java-Applet (schematische Darstellung von Szenario C)

Der „Gatekeeper“ gehört nicht zum Modellierungstool. Er ist Bestandteil von „Visibroker für Java“. Der Gatekeeper dient als Bindeglied (Gateway) zwischen Applets – die CORBA nutzen und während ihrer Ausführung im Web-Browser gewissen Sicherheitsbeschränkungen (Java-Sandbox) unterliegen – und CORBA-Objekten. Der Gatekeeper kann mit Hilfe diverser/umfangreicher Konfigurationsoptionen an verschiedene Einsatzszenarien (u.a. für den Einstatz in Netzwerken, die über Firewalls verbunden sind) angepasst werden. Der Gatekeeper ist auch in der Lage, selbst als Webserver zu fungieren. Von dieser Möglichkeit wurde bei „Szenario C“ Gebrauch gemacht. Nähere Informationen zum Gatekeeper sind in [VBJ2] zu finden.
Szenario A bietet den Vorzug der einfachen Konfiguration des verwendeten Rechners, da lediglich eine Ausführungsumgebung von Java erforderlich ist. Dabei wird das Modellierungstool (komplett) auf einem Windows NT-Rechner ausgeführt. Die Client/Server-Varianten eröffnen die Möglichkeit, den Client-Teil des Modellierungstools auf einem anderen Rechner (der gegebenenfalls nicht unter Windows NT betrieben wird) auszuführen. Falls die Bedienung von TSecMan z.B.

an einem UNIX-Rechner erfolgt, könnte der Client des Modellierungstools ebenfalls auf diesem Rechner ausgeführt werden und so einen aktuellen (online) Überblick über die verwaltete(n) Windows NT-Umgebung(en) ermöglichen.

7.2 Ein kommentiertes Einsatzbeispiel

Im Folgenden soll die Bedienung/Funktionsweise des Modellierungstools anhand eines Einsatzbeispiels veranschaulicht werden. Abbildung 34 zeigt die GUI des Modellierungstools (Client als Applet) nach dem Programmstart.

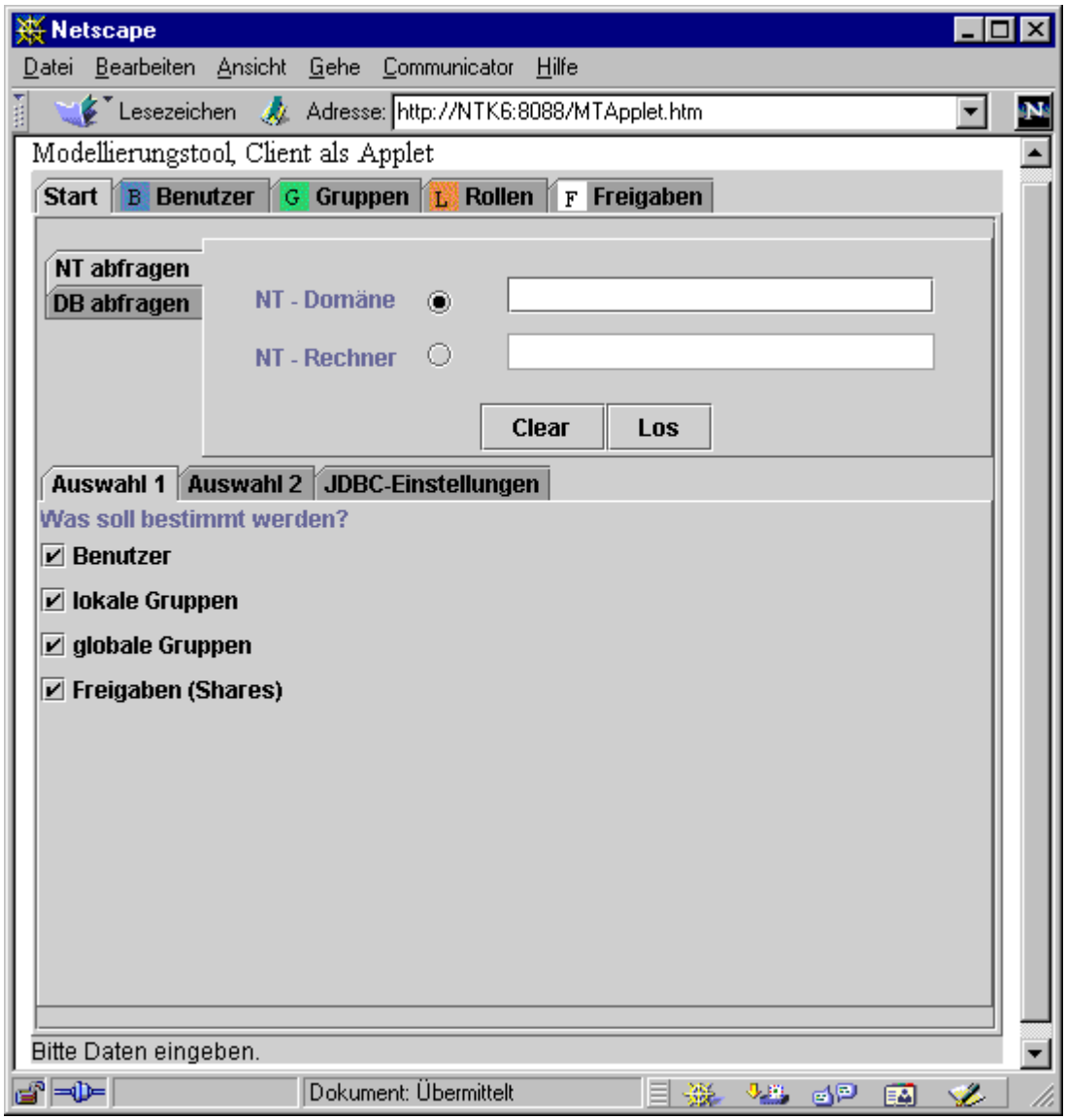

Abbildung 34: Modellierungstool, Ansicht „Start“ mit Registerkarte „NT abfragen“

Die GUI des Modellierungstools wurde mit Hilfe von Registerkarten strukturiert. Die prinzipielle Arbeitsweise des Modellierungstools besteht darin, dass jeweils ein (einzelner) Abfragezyklus ausgelöst wird, der einen NT-Rechner untersucht. Die Registerkarte „Start“ enthält Steuerelemente, um die Ausführung dieses Abfragezyklus zu konfigurieren und zu veranlassen. Nach Abarbeitung des Abfragezyklus werden die gewonnenen Informationen angezeigt. Dazu dienen die Registerkarten „Benutzer“, „Gruppen“, „Rollen“ und „Freigaben“. Im unteren Bildschirmbereich des Modellierungstools befindet sich eine Statuszeile, in der Hinweise oder Fehlermeldungen erscheinen. Unmittelbar nach dem Programmstart ist die Registerkarte „Start“ aktiviert, die im Folgenden auch als Ansicht Start bezeichnet wird. Diese Ansicht enthält mehrere Registerkarten für diverse Optionen des Abfragezyklus. Der Abfragezyklus ist stets „Rechner-orientiert“, d.h. die jeweilige Abfrage bezieht sich auf einen zu untersuchenden Rechnernamen. Die Registerkarte „NT abfragen“ enthält zwei Eingabefelder, in denen der Name einer zu untersuchenden Windows NT-Domäne bzw. der Name eines zu untersuchenden NT-Rechners einzugeben ist. Die Eingabefelder können alternativ angewählt werden. Als Voreinstellung ist der Domänenname angewählt, in diesem Fall, wird am Anfang des Abfragezyklus anhand des Domänennamens der zugehörige PDC bestimmt. Diese Vorgehensweise soll sicherstellen, dass der Domänenname korrekt eingegeben wird. Die direkte Angabe des zu untersuchenden Rechnernamens wurde nur deshalb vorgesehen, um auch Client-Rechner einer NT-Domäne untersuchen zu können. Dabei wird allerdings nicht ermittelt, zu welcher Domäne dieser Rechner gehört.
Das Betätigen der Schaltfläche „Clear“ bewirkt das Löschen der Eingabefelder. Die Schaltfläche „Los“ startet den Abfragezyklus. Welche Informationen zu ermitteln sind, kann in der Registerkarte „Auswahl 1“ festgelegt werden, wobei globale Gruppen nur dann berücksichtigt werden, wenn in der Registerkarte „NT abfragen“ der Domänenname angewählt ist. Bei der Untersuchung von Client-Rechnern wird also generell auf die Ermittlung globaler Gruppen verzichtet.
Für die Bestimmung der Benutzer kann mit Hilfe der Registerkarte „Auswahl 2“, siehe Abbildung 35, spezifiziert werden, welche Typen von Benutzerkonten (vgl. Abschnitt 5.4 Tabelle 7) zu berücksichtigen sind.

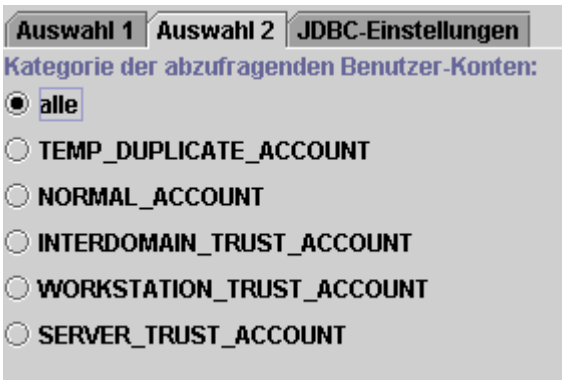

Abbildung 35: Ansicht „Start“, Registerkarte „Auswahl 2“

In Abbildung 35 wurden explizit alle Kontentypen ausgewählt (nach dem Programmstart ist zunächst „NORMAL_ACCOUNT“ als Voreinstellung aktiviert). In dem hier geschilderten Beispiel soll eine Windows NT-Domäne namens „DOM_1“ untersucht werden. Wie in Abbildung 36 zu sehen ist, wurde der Domänenname eingegeben und der Abfragezyklus gestartet (siehe Statuszeile).

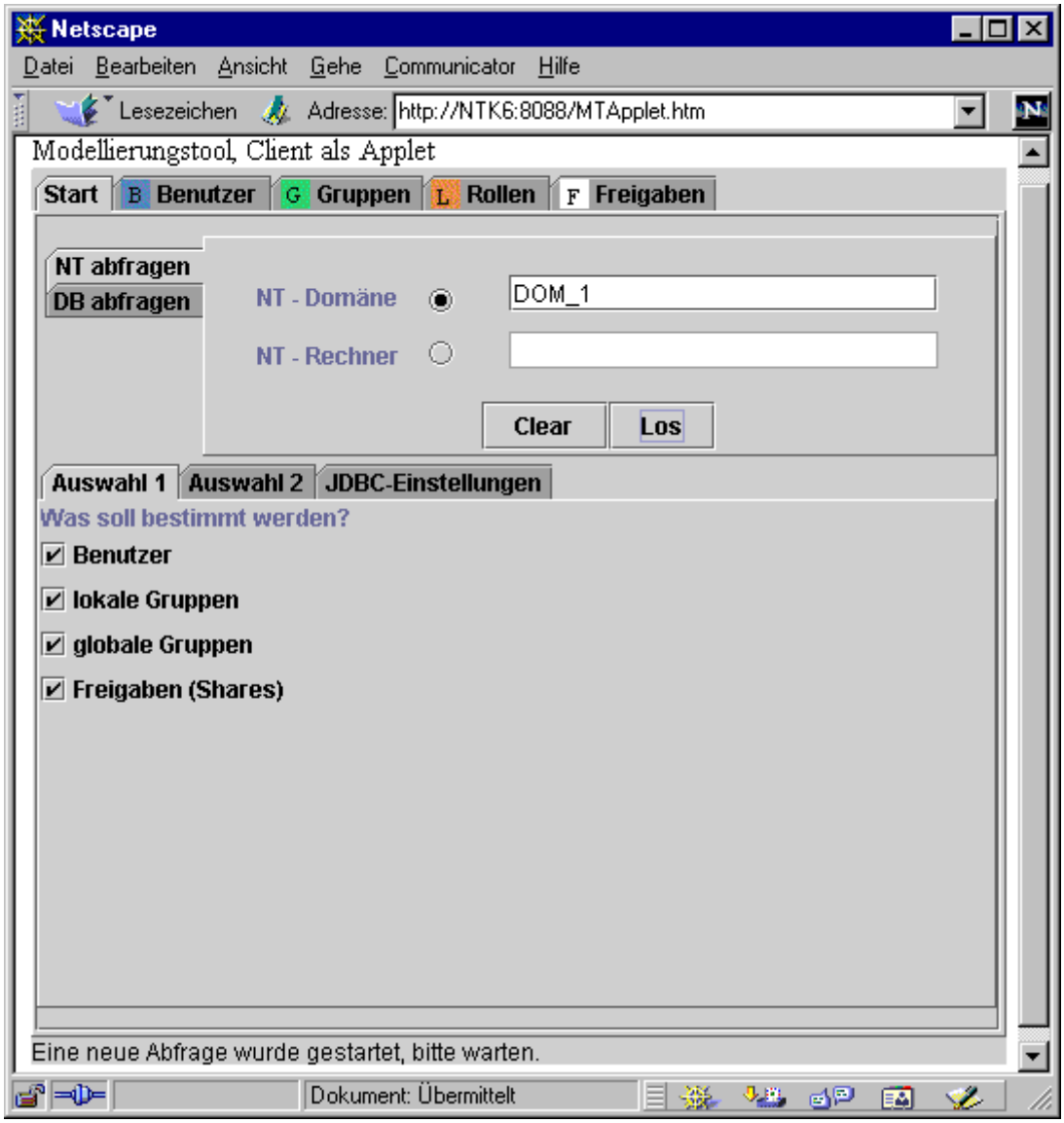

Abbildung 36: Ansicht „Start“ während der Abarbeitung eines Abfragezyklus

Das Ende des Abfragezyklus wird in der Statuszeile mitgeteilt. Abbildung 37 zeigt die Registerkarte „Benutzer“ mit den erhaltenen Informationen.

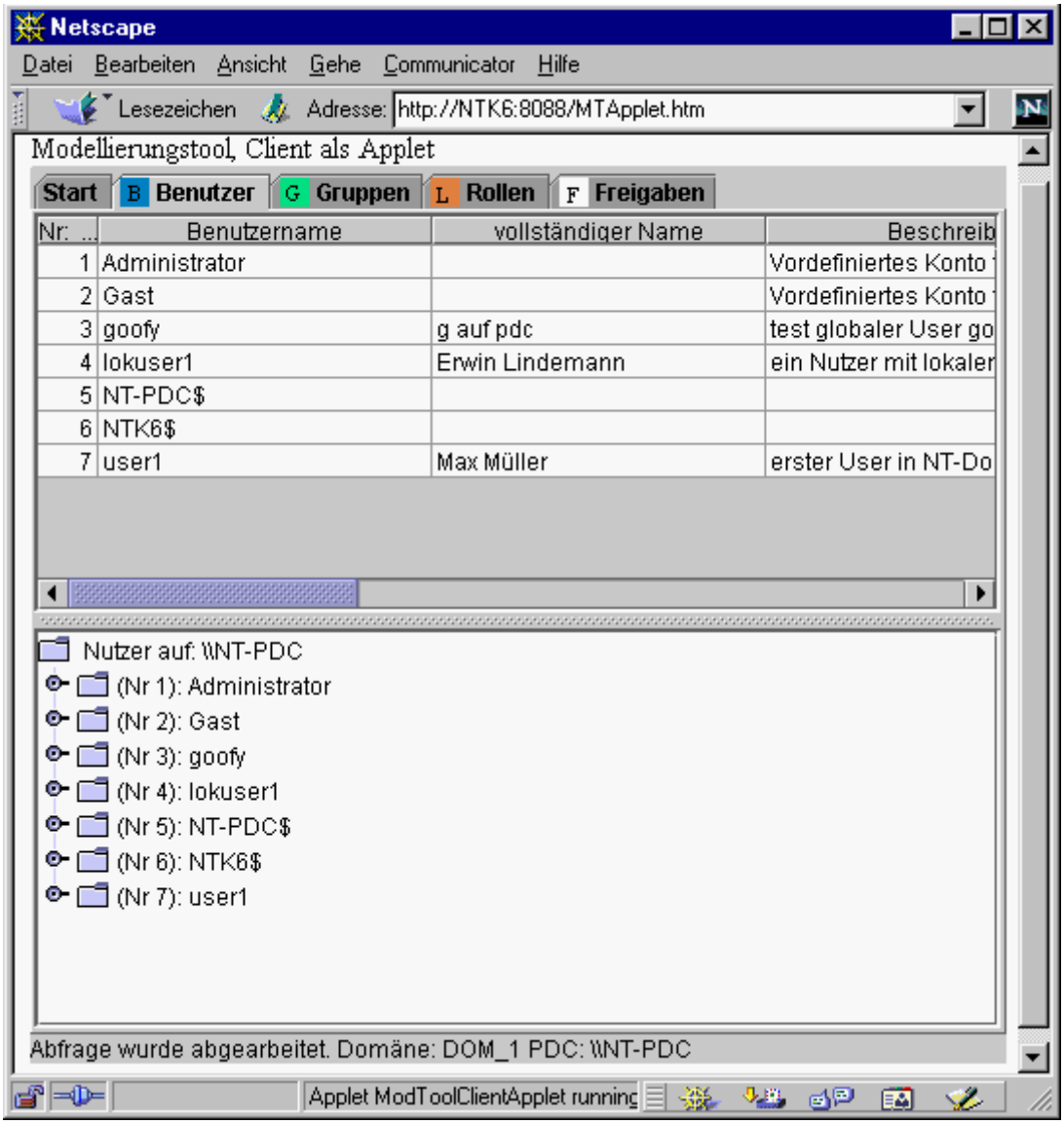

Abbildung 37: Registerkarte „Benutzer“

Die Registerkarte „Benutzer“ enthält in der oberen Hälfte eine Tabelle mit Informationen zu den einzelnen Benutzerkonten. Die angezeigte Nummer in der ersten Tabellenspalte stammt nicht vom jeweiligen Benutzerkonto, sondern soll die Orientierung innerhalb der Registerkarte „Benutzer“ erleichtern. Die untere Hälfte der Registerkarte enthält eine Baumstruktur. Die Trennlinie zwischen Tabelle und Baumstruktur kann verschoben werden. Die Untergliederung in jeweils eine Tabelle und eine Baumstruktur wurde auch in den Registerkarten „Gruppen, „Rollen“ und Freigaben verwendet. In der Registerkarte „Benutzer“ soll mit Hilfe der Baumstruktur, die Gruppenmitgliedschaft der Benutzer veranschaulicht werden, siehe Abbildung 38.

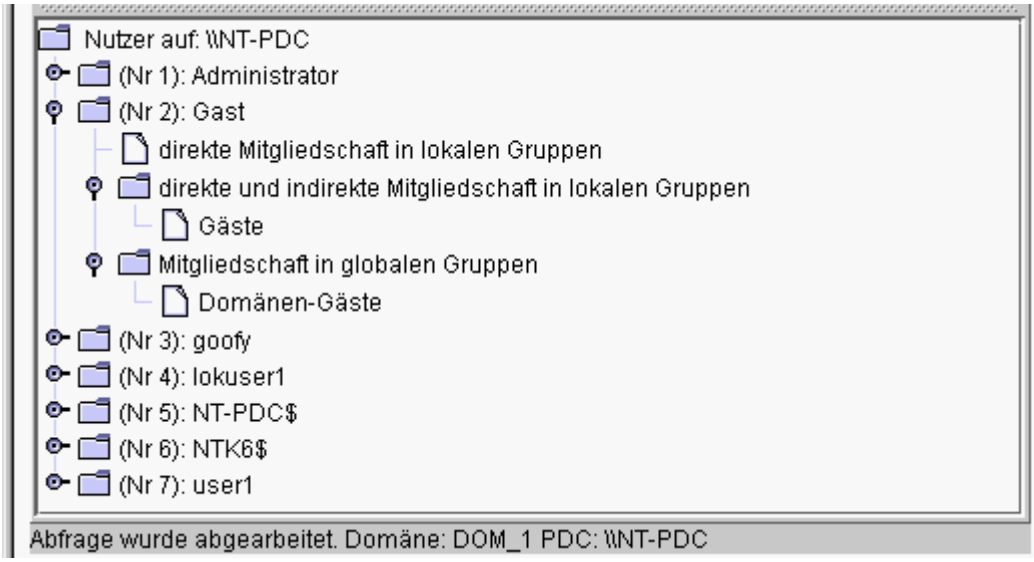

Abbildung 38: Baumstruktur „Benutzer“

Für jeden Benutzer wird die Mitgliedschaft in lokalen Gruppen zunächst ohne und danach mit Berücksichtigung einer indirekten Mitgliedschaft bestimmt. In Abbildung 39 ist die Registerkarte „Gruppen“ dargestellt.

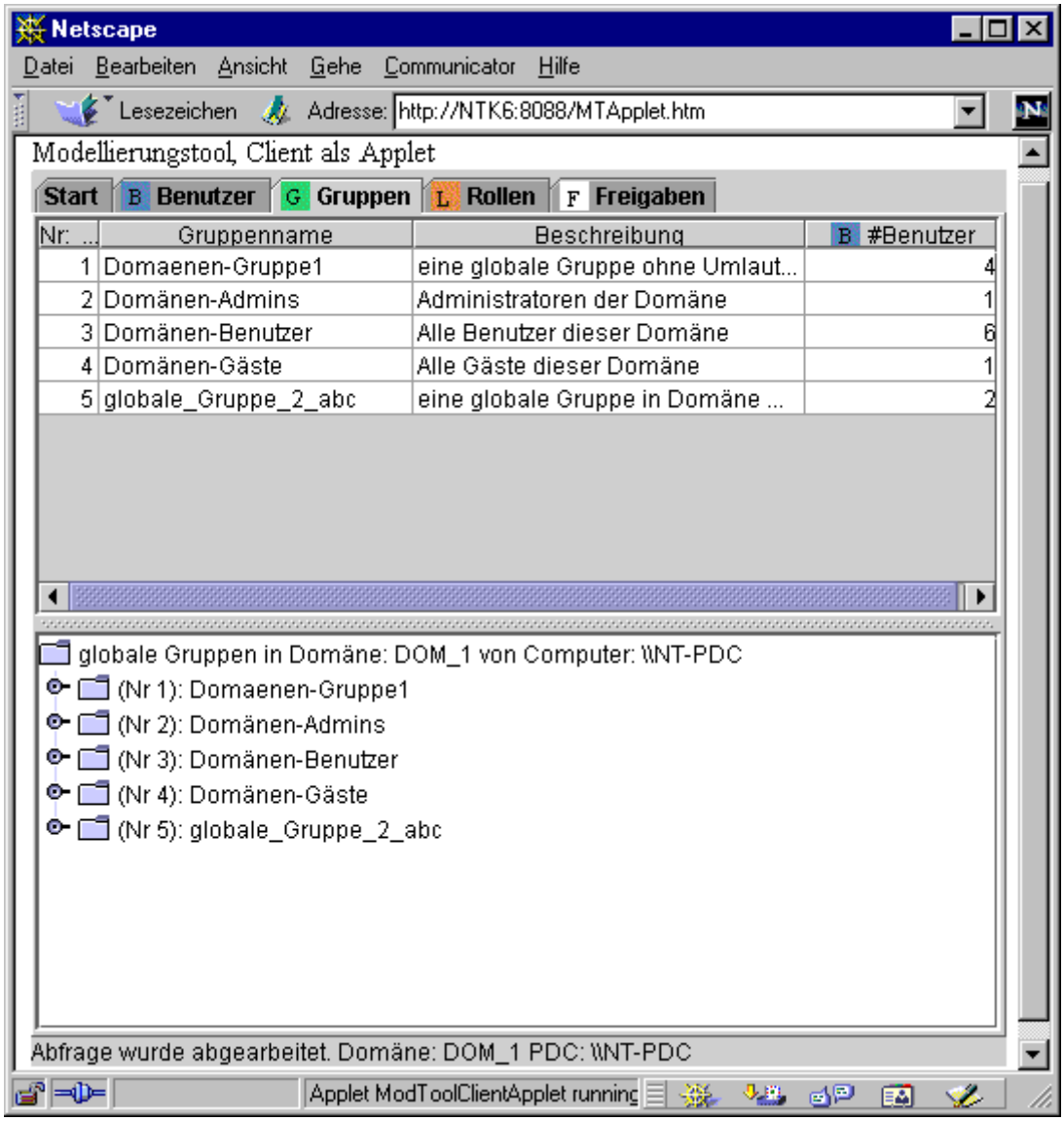

Abbildung 39: Registerkarte „Gruppen“ (globale Gruppen)

Innerhalb dieser Registerkarte werden Informationen über die ermittelten globalen Gruppen dargestellt. Das Modellierungstool bestimmt für jede globale Gruppe deren Mitglieder. Die Tabellenspalte „#Benutzer“ enthält die Anzahl der Benutzer, die in der jeweiligen globalen Gruppe Mitglied sind. In der Baumstruktur werden die entsprechenden Namen der Gruppenmitglieder angezeigt, siehe Abbildung 40.

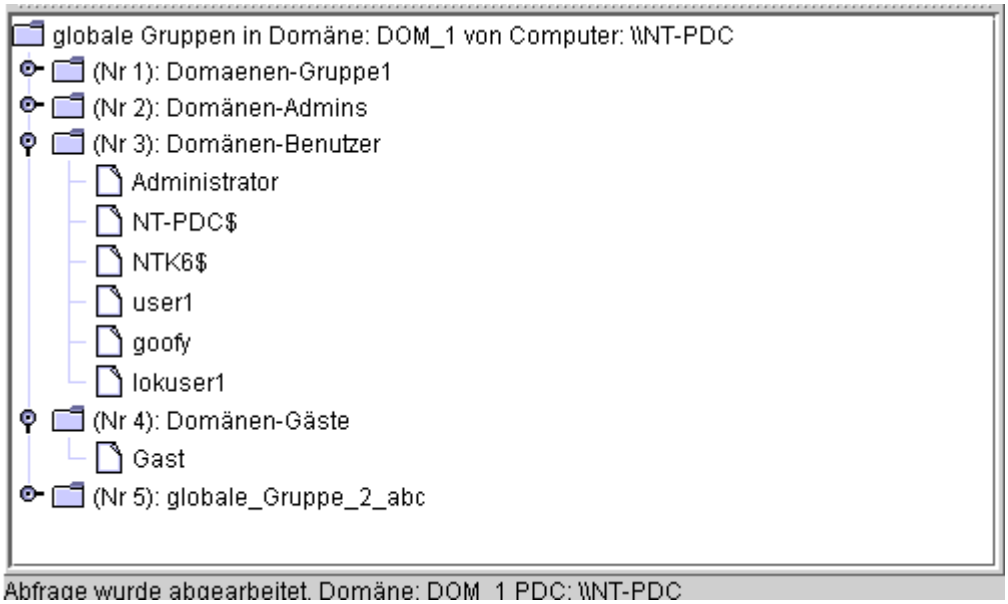

Abbildung 40: Baumstruktur „Gruppen“

Abbildung 41 zeigt die Registerkarte „Rollen“ in der Informationen zu den vorhandenen lokalen Gruppen dargestellt werden.

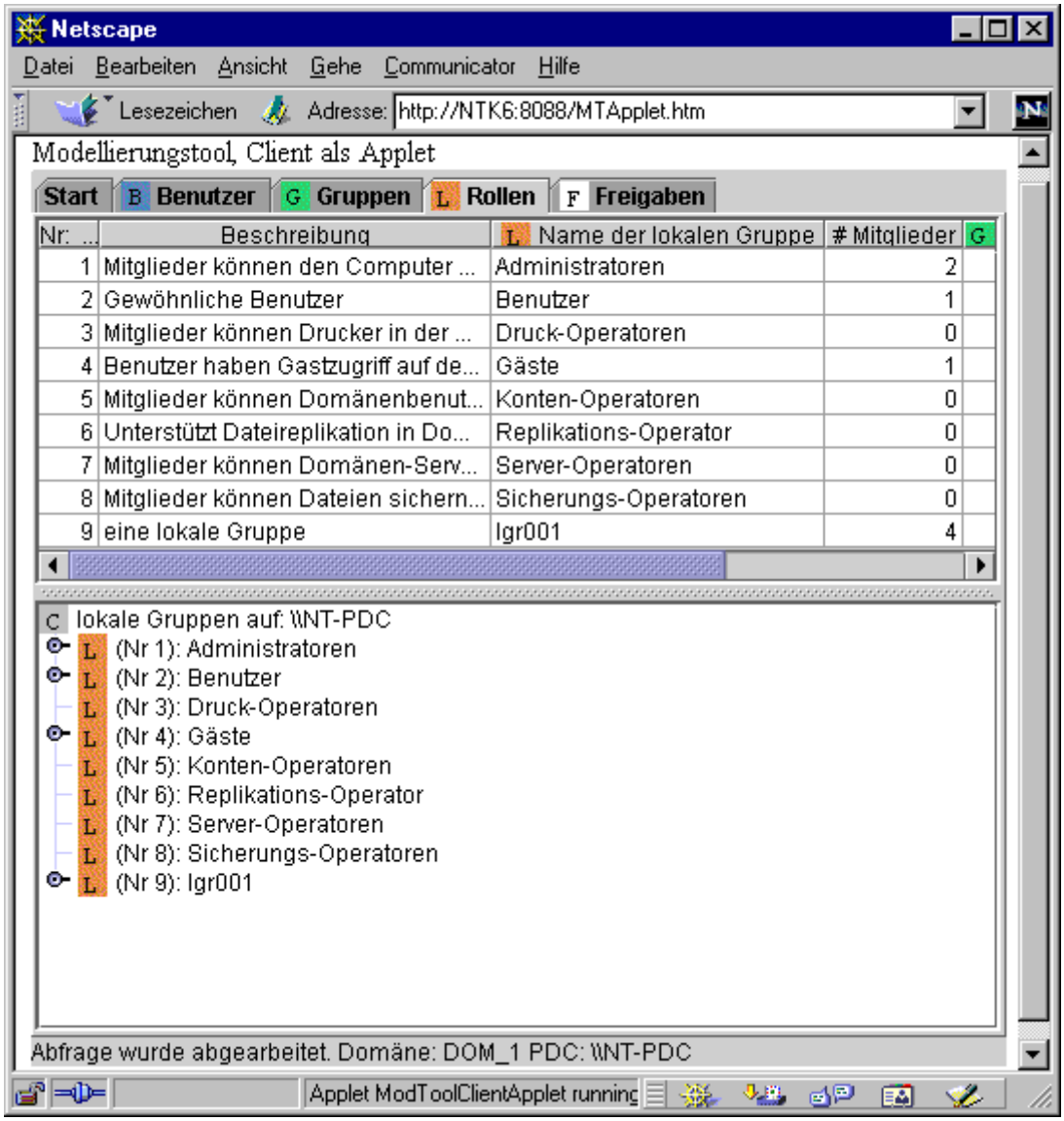

Abbildung 41: Registerkarte „Rollen“ (lokale Gruppen)

Die Reihenfolge der Tabellenspalten innerhalb der Tabellendarstellung kann – durch das Verschieben einzelner Spalten – beliebig verändert werden. Diese Eigenschaft wird durch die „Swing-Klassen“ von Java bereitgestellt. Sie ist auch für die Tabellen in den anderen Registerkarten verfügbar. Für Abbildung 41 wurden die Reihenfolge der beiden Tabellenspalten für den Namen bzw. die Beschreibung der lokalen Gruppe vertauscht, um dies zu demonstrieren.
Das Modellierungstool bestimmt für jede lokale Gruppe deren (direkte) Mitglieder. In der Tabellenspalte „# Mitglieder“ wird die Anzahl der Mitglieder angezeigt. Diese Anzahl wird in den beiden folgenden Spalten nach globalen Gruppen und Benutzerkonten aufgeschlüsselt, siehe Abbildung 42.

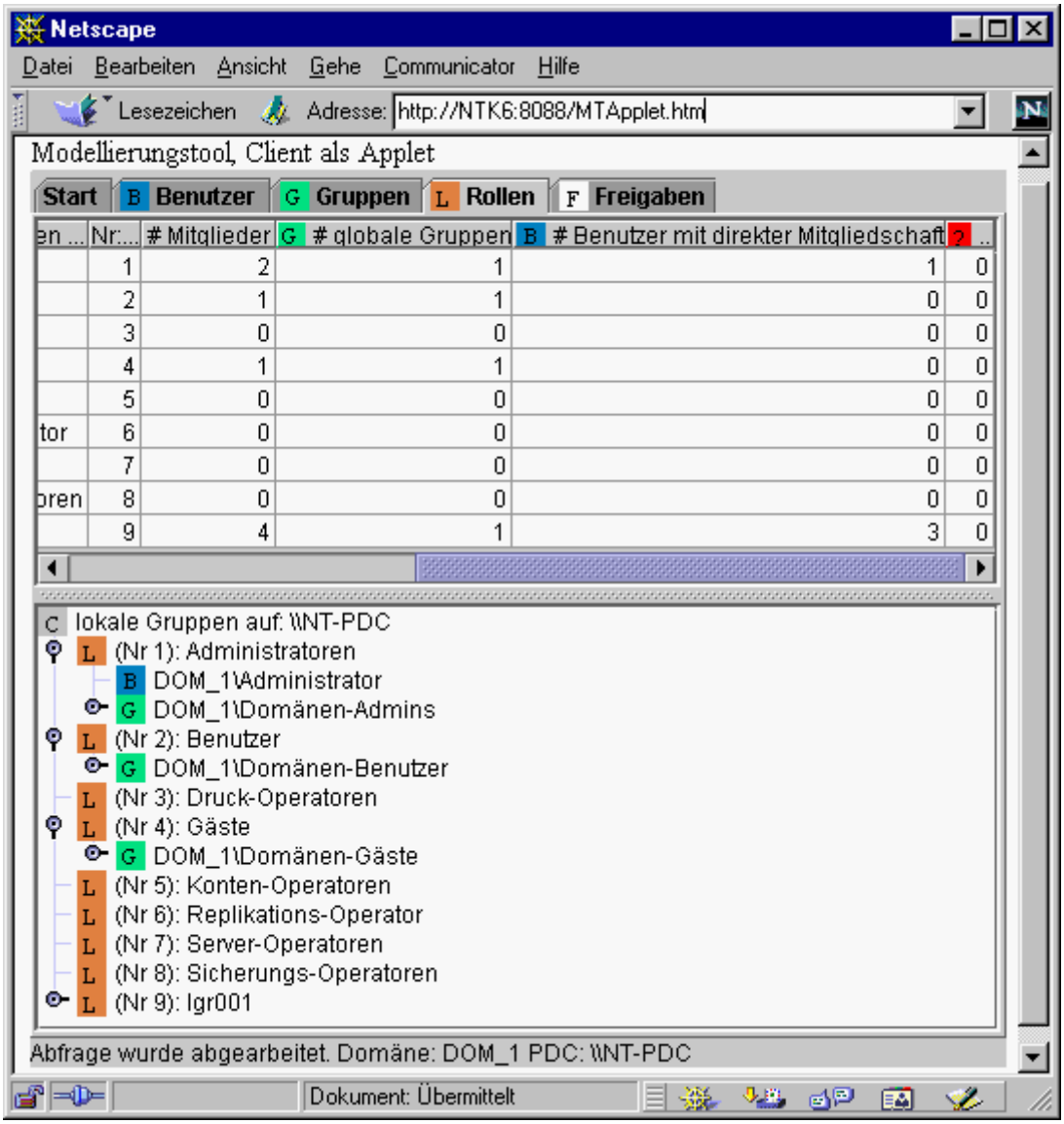

Abbildung 42: Registerkarte „Rollen“ (lokale Gruppen) mit Darstellung direkter Mitglieder

Wie in Abbildung 42 zu sehen ist, wurde für die Baumstruktur zur Darstellung der Mitgliedschaft in lokalen Gruppen eine Anpassung der angezeigten Symbolgrafiken implementiert.
Bei den Mitgliedern der jeweiligen lokalen Gruppen wird auch angegeben, woher das Mitgliedskonto stammt. In diesem Beispiel stammen alle Mitgliedskonten von „DOM_1“, also aus der untersuchten Domäne. Die Mitglieder der enthaltenen globalen Gruppen werden vom Modellierungstool nicht ermittelt. Für globale Gruppen, die zur untersuchten Domäne gehören, werden aber entsprechende Informationen in die Baumstruktur der Registerkarte „Rollen“ übernommen. Falls

also beim Abfragezyklus globale Gruppen zu bestimmen waren, erscheinen die dabei festgestellten Benutzer auch innerhalb der Registerkarte „Rollen“, siehe Abbildung 43.

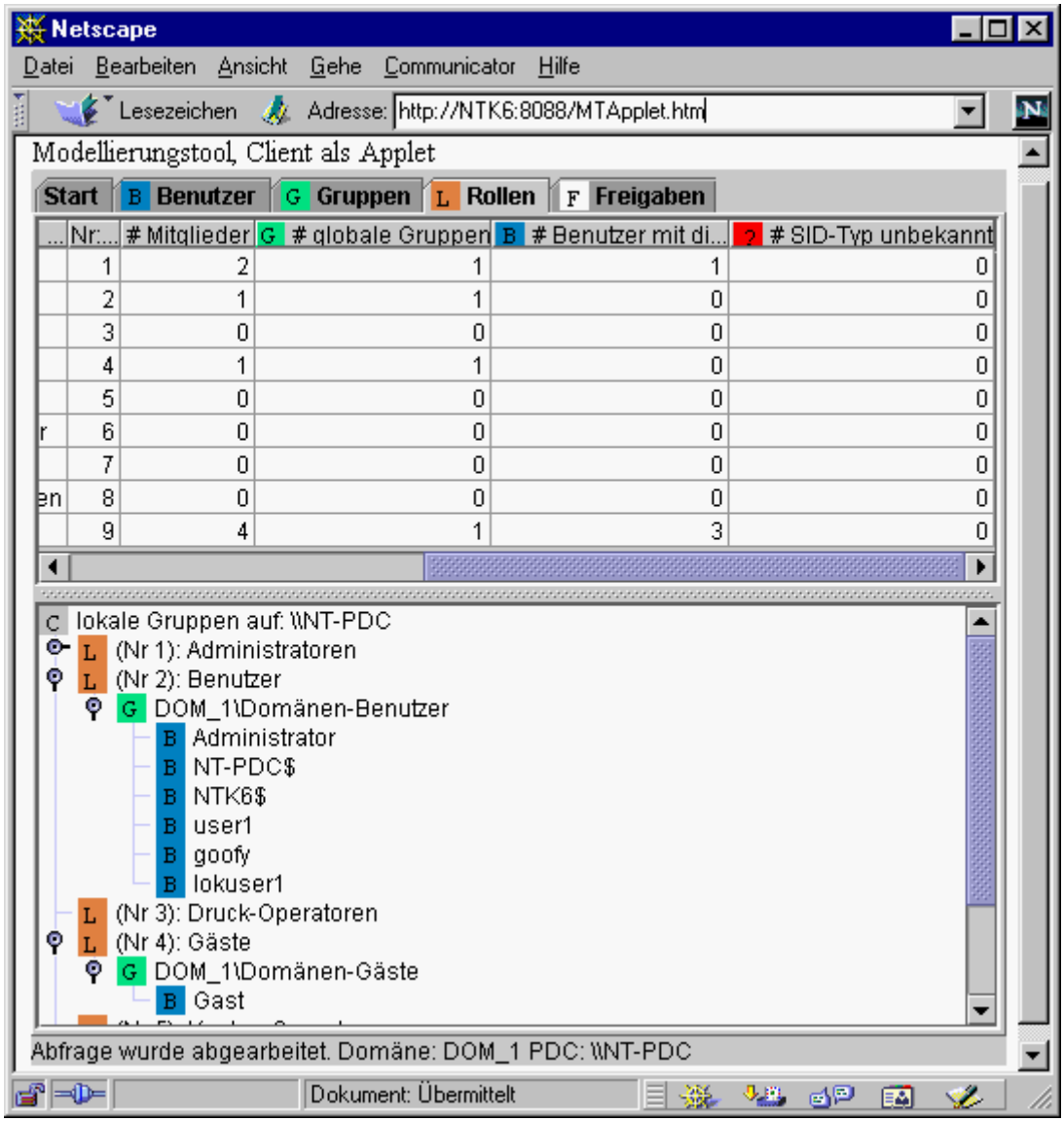

Abbildung 43: Registerkarte „Rollen“ (lokale Gruppen) mit Darstellung indirekter Mitglieder

In Abbildung 43 ist die Tabellenspalte „# SID-Typ unbekannt“ zu erkennen. In dieser Tabellenspalte ist die Anzahl jener Mitglieder wiedergegeben, zu denen keine näheren Informationen verfügbar waren. Dazu kann es beispielsweise kommen, wenn ein Mitglied der betreffenden lokalen Gruppe aus einer anderen Domäne stammt und diese Domäne während des Abfragezyklus nicht erreichbar war. Für solche Mitgliedskonten konnte dann auch nicht bestimmt werden, ob es sich um

einen Benutzer oder eine globale Gruppe handelt. Die Registerkarte „Freigaben“ enthält Informationen zu Freigaben (Shares), siehe Abbildung 44.

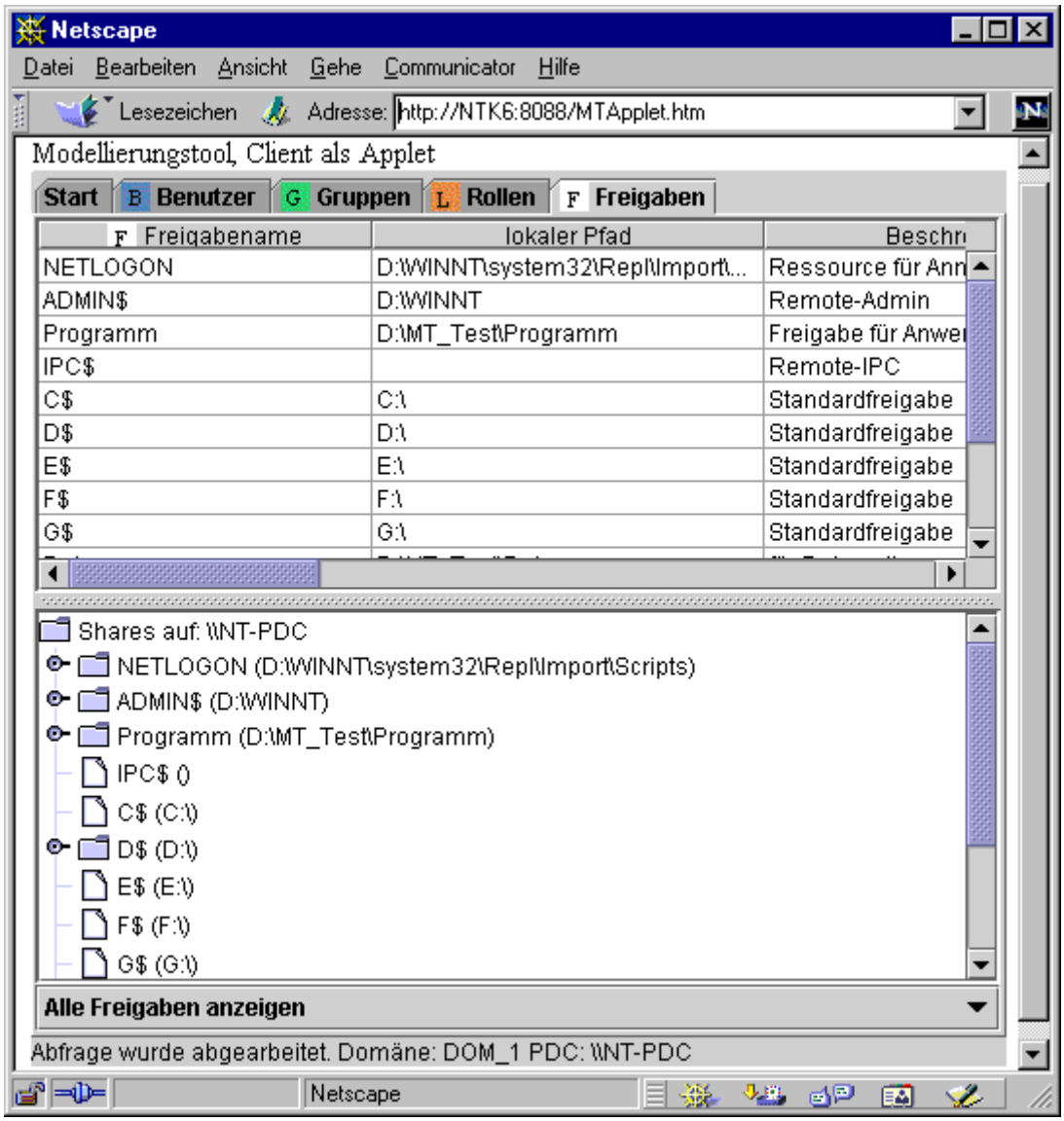

Abbildung 44: Registerkarte „Freigaben“ (Shares)

Das Modellierungstool ermittelt Informationen zu allen vorhandenen Freigaben. Für freigegebene Verzeichnisse werden zusätzlich die Freigabeberechtigungen und die Verzeichnisberechtigungen (NTFS-Berechtigungen) bestimmt. Die Baumstruktur in der Registerkarte „Freigaben“ dient zur Veranschaulichung der ermittelten Zugriffsberechtigungen. Mit dem Auswahlfeld unterhalb der Baumstruktur kann beeinflusst werden, inwieweit administrative Freigaben in der Registerkarte erscheinen. In Abbildung 45 wurde beispielsweise auf die Anzeige administrativer Freigaben verzichtet.

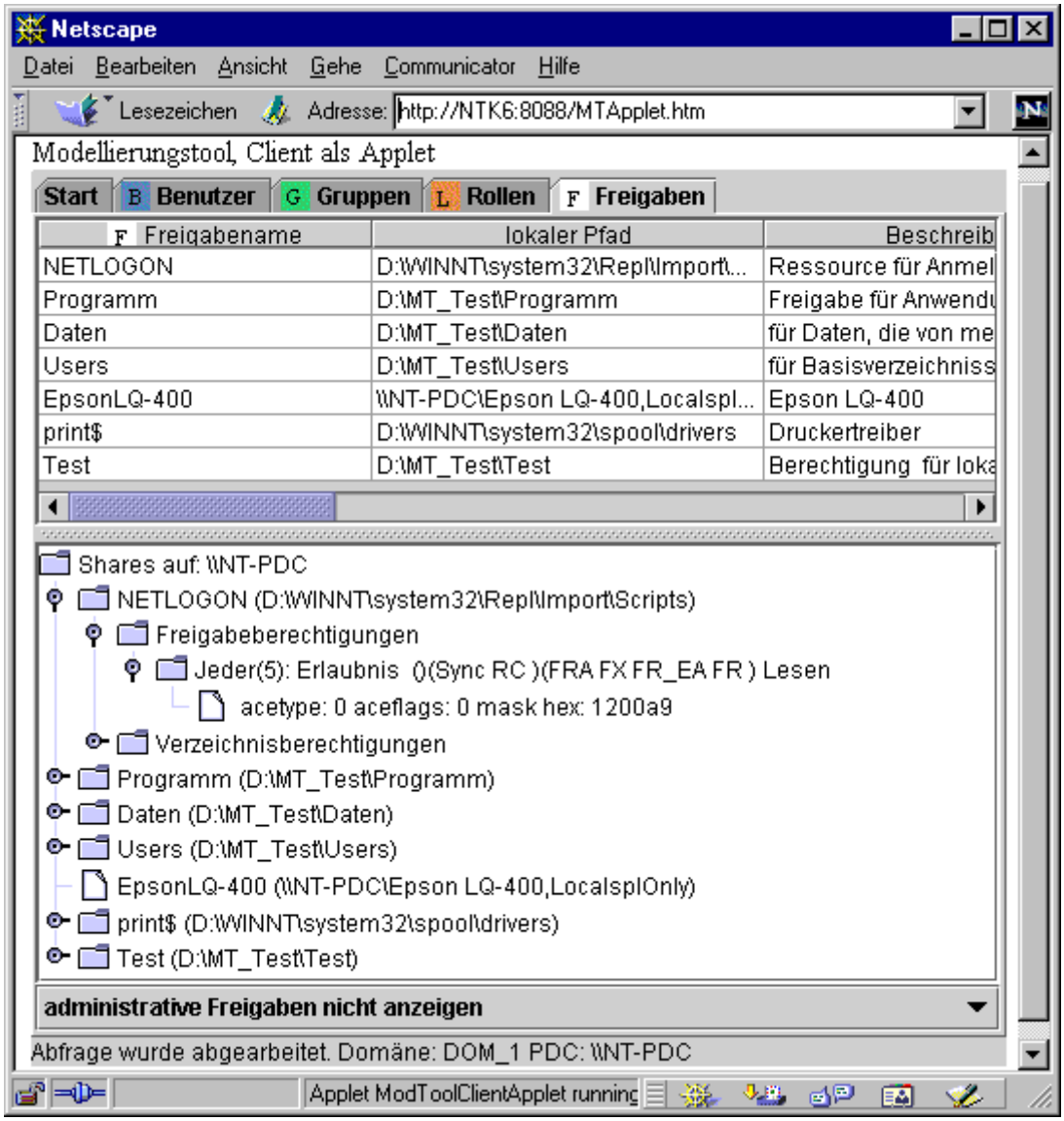

Abbildung 45: Registerkarte „Freigaben“ (Shares), Ansicht mit Freigabeberechtigungen für die Freigabe „NETLOGON“

Zur Darstellung der Zugriffsberechtigungen werden innerhalb der Baumstruktur Informationen zu den einzelnen ACEs angegeben. Wie Abbildung 45 zu entnehmen ist, bestehen die Freigabeberechtigungen für die Freigabe „NETLOGON“ im vorliegenden Beispiel aus einem ACE, welcher der (besonderen) Gruppe „Jeder“ die Freigabeberechtigung „Lesen“ erlaubt. Im Einzelnen handelt es sich um folgende Angaben:

- Name des Trustees: „Jeder“
- eine Nummer, mit der der Kontotyp (SID-Typ) des Trustees gekennzeichnet wird: „5“

- eine Angabe zum ACE-Typ: „Erlaubnis“
- eine Kurzinterpretation der Zugriffsmaske: Diese Kurzinterpretation umfasst die abgekürzten Bezeichnungen der Bits, die in der Zugriffsmaske gesetzt sind sowie die Bezeichnung der Freigabeberechtigung („Lesen“).

Anschließend sind die Zahlenwerte von ACE-Typ, ACE-Flags und Zugriffsmaske angefügt, wobei die Zugriffsmaske als hexadezimale Zahl angegeben wird. Anhang C gibt Auskunft über Einteilung, Bezeichnung und Bedeutung der Bits innerhalb der Zugriffsmaske.
Die vom Modellierungstool verwendeten Nummern zur Unterscheidung der möglichen Kontentypen (SID-Typen) eines Trustees gehen aus Tabelle 16 hervor.

Nummer	Bezeichnung des SID-Typs (laut [MSD98])	Bedeutung
1	SidTypeUser	Benutzer
2	SidTypeGroup	globale Gruppe
3	SidTypeDomain	Domäne (Bemerkung: Dieser SID-Typ sollte in einem ACE nicht vorkommen.)
4	SidTypeAlias	lokale Gruppe
5	SidTypeWellKnownGroup	besondere (implizite) Gruppe
6	SidTypeDeletedAccount	gelöschtes Konto
7	SidTypeInvalid	SID ist ungültig
8	SidTypeUnknown	unbekannter SID-Typ
-2		Es konnten keine Informationen über den Trustee ermittelt werden.

Tabelle 16: die vom Modellierungstool verwendeten Nummern zur Kennzeichnung des Kontotyps (SID-Typs) eines Trustees

Die Verzeichnisberechtigunen werden innerhalb der Baumstruktur der Registerkarte „Freigaben“ in ähnlicher Form wie die Freigabeberechtigungen dargestellt, siehe Abbildung 46.

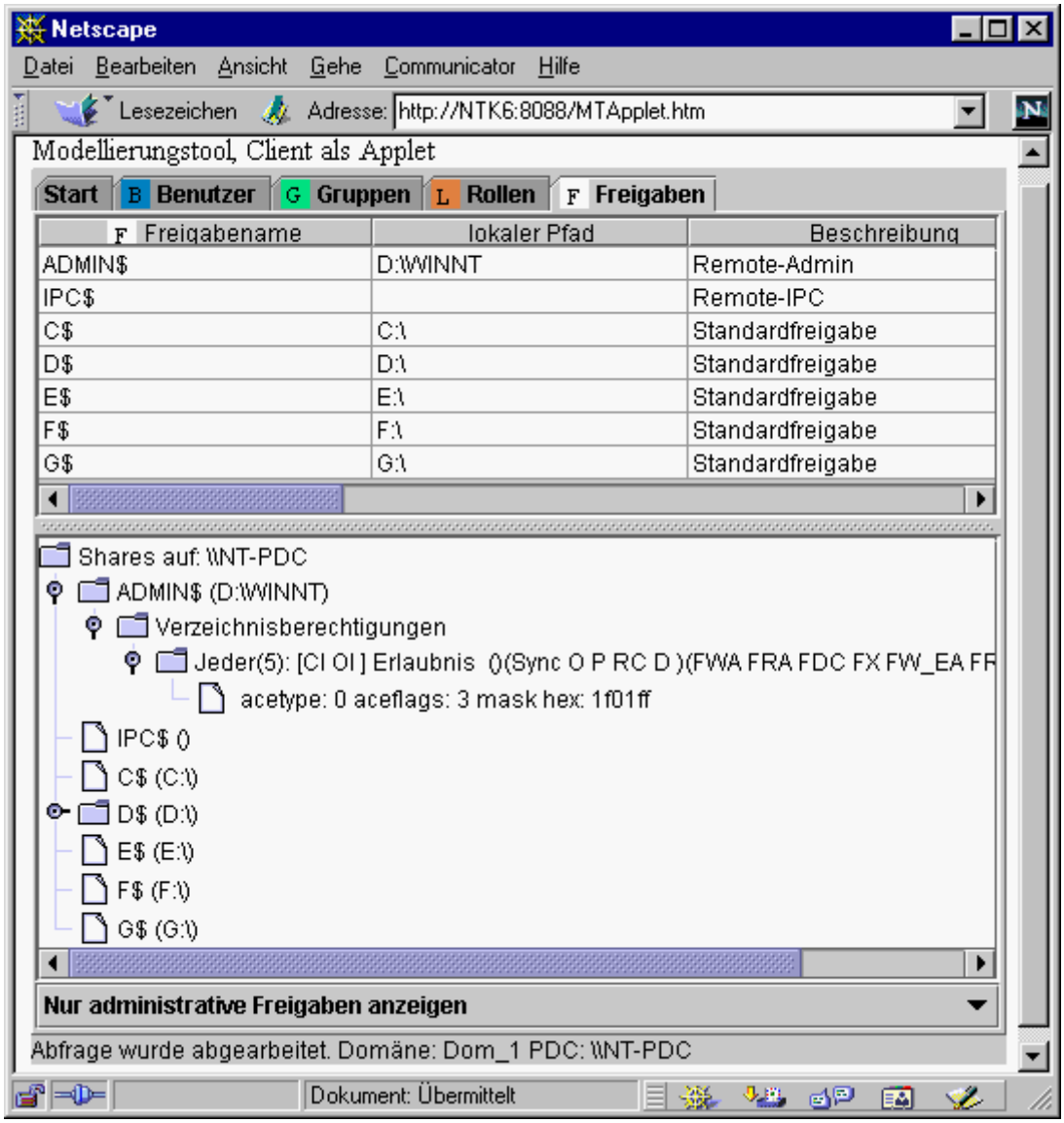

Abbildung 46: Registerkarte „Freigaben“ (Shares), Ansicht mit Verzeichnisberechtigungen für die Freigabe „ADMIN$“

In Abbildung 46 werden ausschließlich administrative Freigaben gezeigt. Für Verzeichnisberechtigungen besteht die Kurzinterpretation der Zugriffsmaske allerdings nur aus den Bezeichnungen der gesetzten Bits. Zusätzlich ist bei den Informationen zu den Verzeichnisberechtigungen (hinter dem Kontotyp des Trustees) in eckigen Klammern eine Aufzählung der gesetzten ACE-Flags enthalten. Die dabei verwendeten Abkürzungen sind in Tabelle 17 aufgeführt und erläutert.

Bezeichnung des ACE-Flags (laut [MSD98])	Abkürzung (Zahlenwert)	Bedeutung
CONTAINER_INHERIT_ACE	CI (2)	Der ACE wird an Verzeichnisse vererbt.
INHERIT_ONLY_ACE	IO (8)	Dieser ACE gilt nur für die (im Verzeichnis) enthaltenen Objekte. Der ACE gilt nicht für das Verzeichnis selbst.
OBJECT_INHERIT_ACE	OI (1)	Der ACE wird an Dateien vererbt.

Tabelle 17: Übersicht zu Bezeichnung und Bedeutung der ACE-Flags im Rahmen von Verzeichnisberechtigungen

Die vom Modellierungstool ermittelten Informationen, werden im Folgenden als „Modell“ bezeichnet. Das Speichern des Modells wird innerhalb der Ansicht „Start“ veranlasst, siehe Abbildung 47.

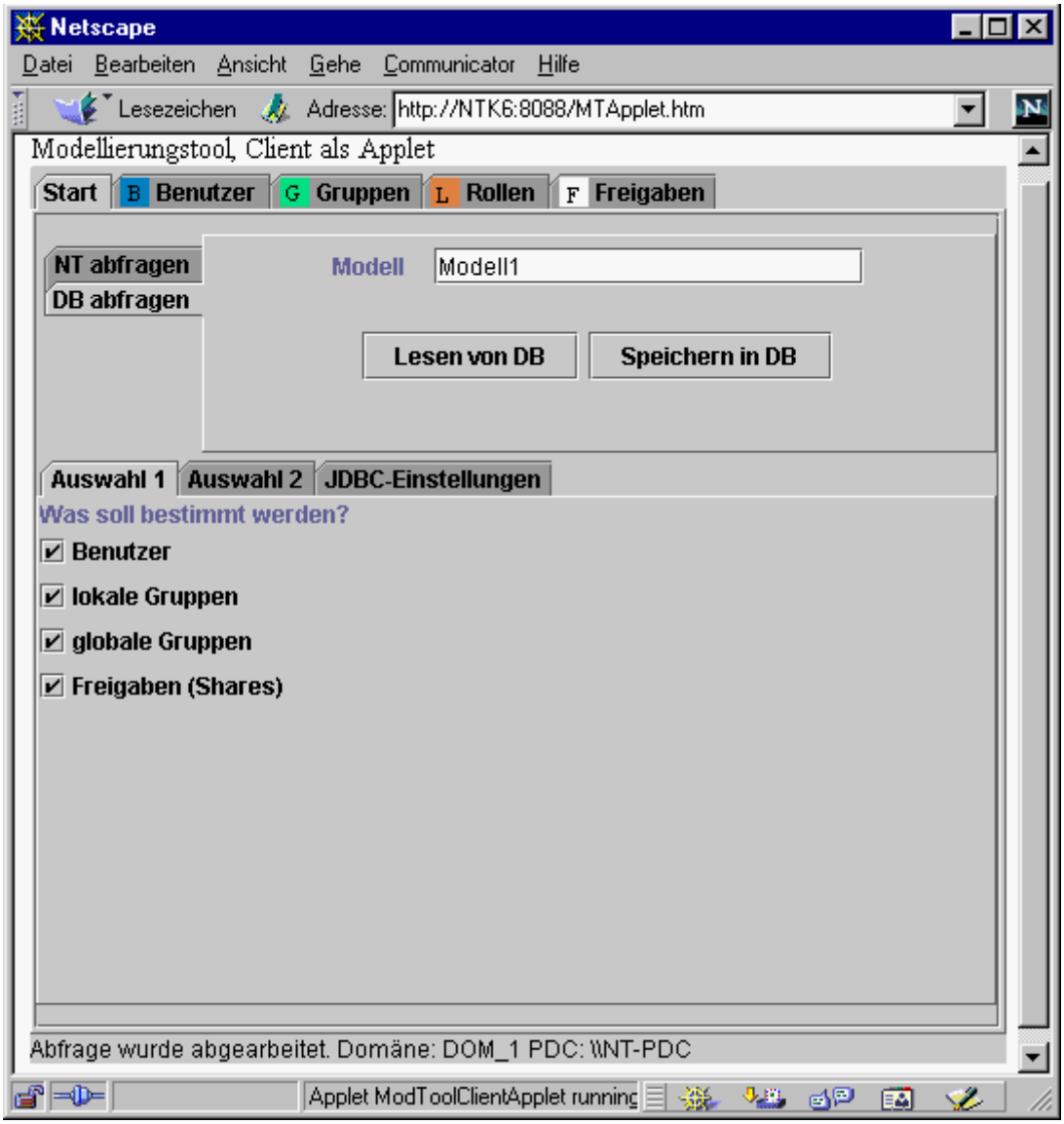

Abbildung 47: Ansicht „Start“ (Modell in Datenbank speichern) mit Registerkarte „Auswahl 1“

Die Registerkarte „DB abfragen“ in der Ansicht „Start“ enthält ein Eingabefeld für die Bezeichnung des Modells. Wie Abbildung 47 zeigt, wurde in diesem Beispiel „Modell1“ als Modellname verwendet. Optional kann mit Hilfe der Registerkarte „Auswahl 1“ der Umfang der zu speichernden Informationen eingeschränkt werden. (Die Registerkarte „Auswahl 2“ wird nicht ausgewertet.)
Die Registerkarte „JDBC-Einstellungen“ bietet Konfigurationsoptionen für die Verbindung zur Datenbank, siehe Abbildung 48.

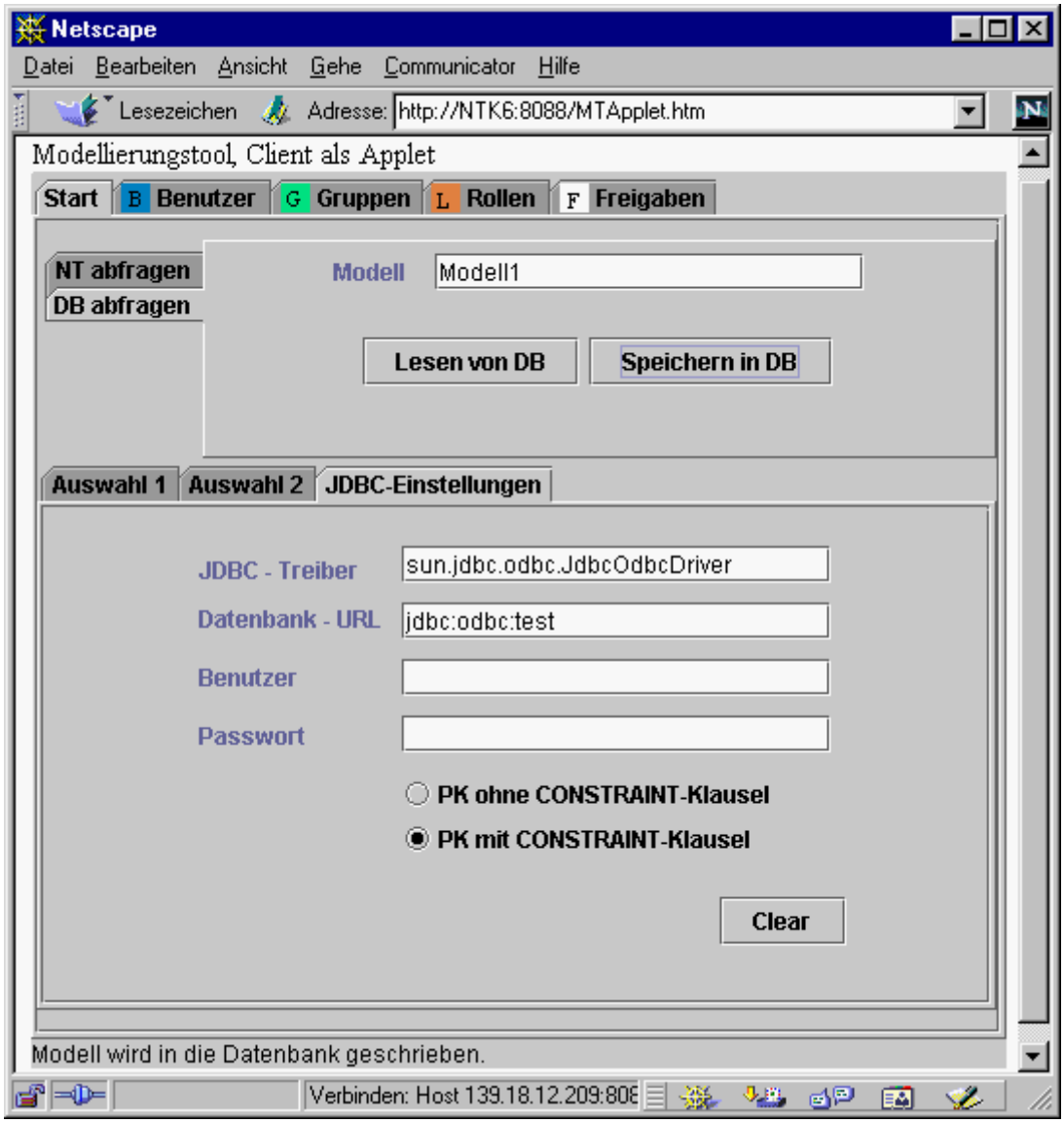

Abbildung 48: Ansicht „Start“ (Modell in Datenbank speichern) während des Speicherns

Die Eingabefelder der Registerkarte „JDBC-Einstellungen“ dienen zur Festlegung der zu nutzenden Datenbankverbindung. Die Schaltfläche „Clear“ löscht alle Eingabefelder der Registerkarte „JDBC-Einstellungen“. Das Speichern des Modells wird in der Registerkarte „DB abfragen“ durch Betätigen der entsprechenden Schaltfläche („Speichern in DB“) gestartet. Der JDBC-Teil des Modellierungstools wählt den angegebenen JDBC-Treiber (hier: „sun.jdbc.odbc.JdbcOdbcDriver“), um eine Verbindung zur angegebenen Datenbank (hier: „jdbc:odbc:test“) herzustellen. Für nähere Details zur Syntax und Semantik dieser Angaben sei auf [Ham99], [SUN1] sowie die Dokumentation des jeweils eingesetzten JDBC-Treibers

verwiesen. (Als Voreinstellung wird vom Modellierungstool vorgeschlagen, mit Hilfe des bei Java enthaltenen JDBC-ODBC-Brückentreibers eine ODBC-Verbindung namens „test“ zu verwenden.)
Bei „Benutzer“ und „Passwort“ können die für die Anmeldung an die gewählte Datenbank benötigten Informationen eingegeben werden. Für die obige Datenbankverbindung waren diese Angaben nicht erforderlich.
Nachdem das Modell gespeichert wurde, erscheint eine entsprechende Meldung in der Statuszeile, siehe Abbildung 49.

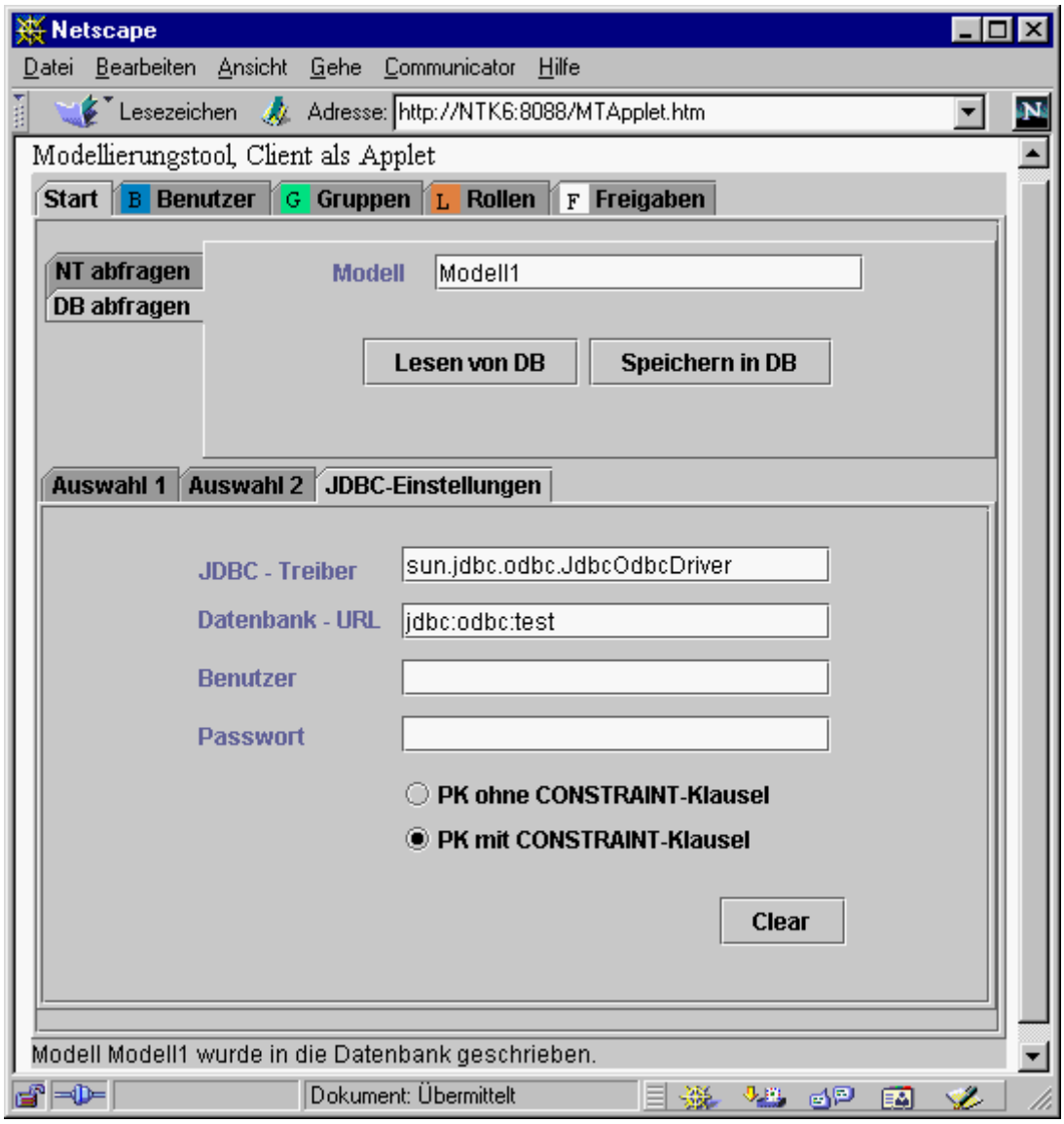

Abbildung 49: Registerkarte „Start“ (Modell in Datenbank speichern) nach der Speicherung

Bei der Entwicklung des Modellierungstools wurde versucht, den JDBC-Teil so zu

implementieren, dass das Modellierungstool mit unterschiedlichen Datenbanken zusammenarbeiten kann. Dies ist durch JDBC nicht automatisch gewährleistet, da Datenbanken verschiedener Hersteller im Allgemeinen herstellerspezifische „Eigenheiten“ bei der Unterstützung von SQL (Structured Query Language) aufweisen. (SQL ist die Standardsprache zum Zugriff auf relationale Datenbanken.) Dies betrifft beispielsweise den Umfang der unterstützten SQL-Datentypen oder auch die Syntax von SQL-Anweisungen. Die verbleibende Auswahlmöglichkeit in der Registerkarte „JDBC-Einstellungen“ bezieht sich auf so eine Eigenheit. Dabei kann gewählt werden, mit welcher Syntax die SQL-Anweisung zum Erstellen einer Tabelle (in der Datenbank) den Primärschlüssel (primary key) definiert, d.h. ob eine solche Definition mit „CONSTRAINT“ eingeleitet wird oder nicht.
Das Modellierungstool verwendete folgende SQL-Datentypen:

- VARCHAR
- INTEGER
- SMALLINT

Der Datentyp SMALLINT wird dabei – entsprechend einer Empfehlung aus [Ham99] – anstelle des Datentyps BIT verwendet, der nicht von allen Datenbanken unterstützt wird. Ein anderes Problem betrifft die Strings, die vom Modellierungstool alle einheitlich als SQL-Datentyp VARCHAR (d.h. als Zeichenkette variabler Länge) gespeichert werden. In [Ham99] wird angegeben, dass zumindest eine Länge von 254 Zeichen für VARCHAR von wichtigen Datenbanken unterstützt wird. Das Modellierungstool verwendet 255 Zeichen als Begrenzung der maximalen Länge für Zeichenketten vom SQL-Datentyp VARCHAR.
Namen für lokale Gruppen bei Windows NT können allerdings maximal bis zu 256 Zeichen lang sein, auf eine Berücksichtigung dieses Extremfalls wurde bei der Implementierung des JDBC-Teils verzichtet.
Das Modellierungstool wurde mit folgenden Datenbanken getestet:

- Microsoft Access 97: Die genutzten JDBC-Einstellungen gehen aus Abbildung 49 hervor.
- Sybase SQL Anywhere 5.0: Die genutzten JDBC-Einstellungen zeigt Abbildung 50.
- Oracle8i (8.1.7): Die genutzten JDBC-Einstellungen sind in Abbildung 51 dargestellt.

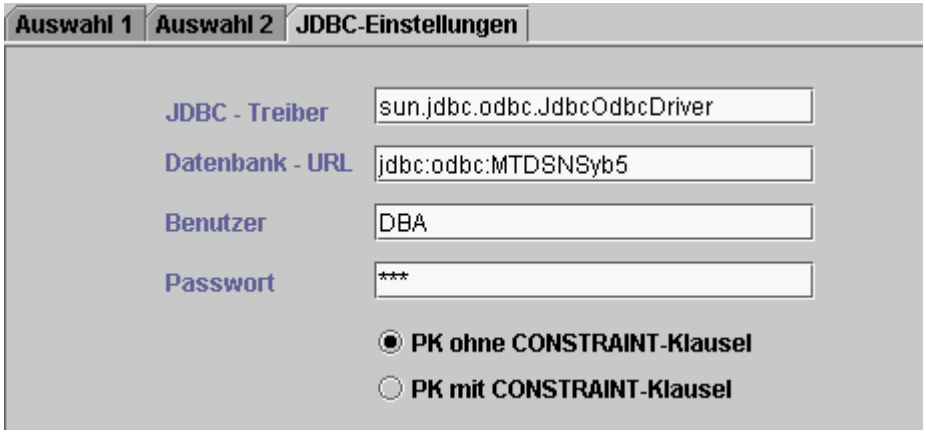

Abbildung 50: Registerkarte „JDBC-Einstellungen“, verwendete Einstellungen beim Zugriff auf Sybase SQL Anywhere 5.0

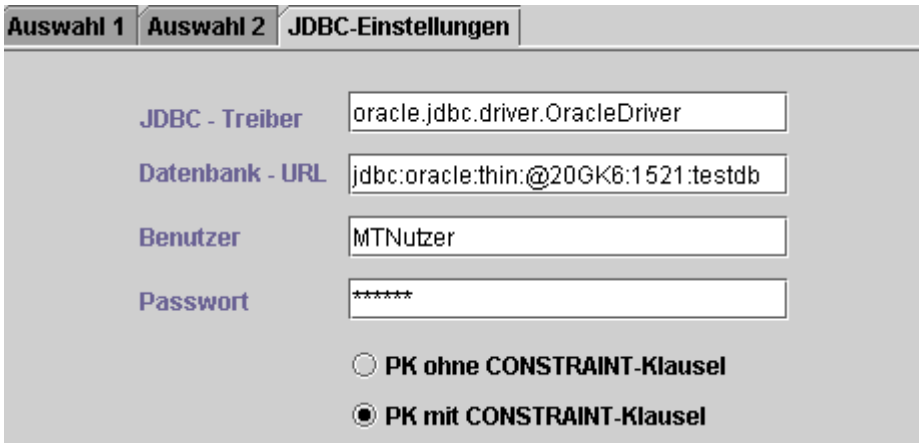

Abbildung 51: Registerkarte „JDBC-Einstellungen“, verwendete Einstellungen beim Zugriff auf Oracle8i (8.1.7)

Zur Ausführung des Modellierungstools sollte die genutzte Java-Umgebung zumindest den Funktionsumfang von JDK1.1 (mit Swing-Klassen) bieten.

8 Zusammenfassung und Ausblick

In vielen Unternehmen (auch Behörden etc.) ist es nötig, den Mitarbeitern einen unkomplizierten, benutzerfreundlichen Zugang zu einer Fülle von Informationen zur Verfügung zu stellen, um ein effizientes Arbeiten zu ermöglichen. Die Erstellung einer maßgeschneiderten Konfiguration der Zugangsmöglichkeiten der Mitarbeiter zur DV-Technik des Unternehmens stellt eine Herausforderung an die jeweiligen Systemverantwortlichen dar. Dabei müssen Zugriffsberechtigungen administriert werden, die festlegen, wie die Mitarbeiter auf IT-Ressourcen zugreifen dürfen. Im Idealfall hat jeder Mitarbeiter (d.h. jeder Benutzer) nur Zugang zu Daten, die er tatsächlich für seine Arbeit benötigt und besitzt dabei nur minimale, d.h. nur die wirklich erforderlichen Zugriffsberechtigungen.
Doch lässt sich dieser Idealfall erreichen?
Nun, welche Daten und Ressourcen wie zu schützen sind und welcher Aufwand dafür betrieben werden darf bzw. muss, hängt vom konkreten Einsatzfall ab. Diesbezüglich sollten alle grundlegenden Richtlinien und Anforderungen eines konkreten Unternehmens als sog. „Sicherheitspolitik“ dieses Unternehmens zusammengefasst und explizit formuliert werden. Das Sicherheitsmanagement dient der Umsetzung einer solchen Sicherheitspolitik.
Bei der Administration der Zugriffsberechtigungen ist die Klärung der beiden folgenden Fragen von zentraler Bedeutung:

- Welche Zugriffsberechtigungen besitzt ein Mitarbeiter?
- Welche Zugriffsberechtigungen benötigt ein Mitarbeiter?

Im Rahmen des Sicherheitsmanagements einer heterogenen IT-Umgebung müssen unterschiedliche Sicherheitsmechanismen gemanagt werden. Dabei gilt es, die „individuellen“, das heißt inkompatiblen, Sicherheitssysteme der eingesetzten Anwendungsprogramme und Betriebssysteme zu steuern. An dieser Stelle ist der Einsatz von Systemmanagementwerkzeugen unerlässlich.
Die Auswahl der einzusetzenden Werkzeuge hängt ebenfalls vom konkreten Einsatzfall ab. Insbesondere ist die jeweils benötigte Integration zwischen den auszuwählenden Werkzeugen, der zu verwaltenden Hard- und Software sowie den anderen NSM-Produkten (die ebenfalls verwendet werden bzw. verwendet werden sollen) zu berücksichtigen.
Mit Hilfe von „TME 10 Security Management“ (TSecMan) kann die Verwaltung der Zugriffsberechtigungen in heterogenen IT-Umgebungen vereinheitlicht werden. Um dies zu erreichen, unterhält TSecMan ein plattformneutrales Sicherheitsmodell (das Rollenmodell), welches auf die Sicherheitssysteme der verwalteten Betriebssysteme abgebildet wird. Zuvor muss dieses Rollenmodell allerdings entsprechend den

Bedürfnissen des jeweiligen Unternehmens erarbeitet werden. Anschließend kann es dann in TSecMan implementiert werden. Durch das Rollenmodell wird eine strukturierte Vorgehensweise bei der Administration von Zugriffsberechtigungen forciert.
Die Stärke von TSecMan liegt in seiner Fähigkeit, Zugriffsberechtigungen – über verschiedenen Plattformen hinweg – in einer einheitlichen Art und Weise, zentral administrieren zu können. Ein sorgfältiger Entwurf des Rollenmodells bildet die Voraussetzung für ein – im Sinne des Sicherheitsmanagements – erfolgreiches Arbeiten mit TSecMan. Einen Teilschritt bei der Einführung von TSecMan in eine bestehende IT-Umgebung bildet die Analyse der bereits vorhandenen (d.h. tatsächlich implementierten) Sicherheitsmaßnahmen. Dazu ist es erforderlich, den aktuellen Zustand der eingerichteten Zugriffsberechtigungen (also Konfigurationsinformationen der vorhandenen Systeme) zu ermitteln und zu dokumentieren. Zumindest in Bezug auf Windows NT bietet TSecMan keine ausreichende Unterstützung zur Lösung dieser Aufgaben.
Das im Rahmen dieser Arbeit entwickelte Modellierungstool dient zur automatisierten Erfassung von solchen Konfigurationsinformationen. Dabei ist das Modellierungstool ausschließlich auf Windows NT ausgerichtet. Speziell geht es um Informationen zu Zugriffsberechtigungen auf Verzeichnisse, die unter Windows NT für netzwerkseitige Zugriffe freigegeben wurden. Da diese Zugriffsberechtigungen prinzipiell sowohl für Gruppen als auch für Benutzer erteilt werden können, ist in diesem Zusammenhang auch die Berücksichtigung der Gruppenmitgliedschaften der Benutzer erforderlich. Das Modellierungstool ist in der Lage, Informationen über vorhandene

- Benutzer,
- lokale und globale Gruppen,
- freigegebene Verzeichnisse (inklusive der Zugriffsberechtigungen)

zu ermitteln, grafisch darzustellen und zu speichern. Dabei werden die Freigabeberechtigungen und die NTFS-Berechtigungen der freigegebenen Verzeichnisse erfasst. Das Modellierungstool ermöglicht die Speicherung der gewonnenen Informationen in einer externen Datenbank. Damit ergeben sich verbesserte Möglichkeiten (beispielsweise für das Suchen und Auswählen von Informationen) bei der Verarbeitung NSM-relevanter Daten. Insgesamt bietet das Modellierungstool folgende Vorteile:

- Die Konfiguration der verwalteten NT-Rechner wird nicht beeinflusst.
- Das Modellierungstool ist unabhängig von anderen NSM-Produkten (und deren Herstellern).
- Es ist nicht an eine spezielle Datenbank gebunden.
- Die gespeicherten Informationen stehen für die Verarbeitung durch weitere Werkzeuge zur Verfügung.

Für ein plattformübergreifendes Management der Zugriffsberechtigungen wurde damit ein Beitrag geleistet.
Die vorliegende Arbeit soll auch zur Durchführung zusätzlicher Untersuchungen und zur Entwicklung weiterer Anwendungen auf dem Gebiet des Sicherheitsmanagements anregen. So bieten insbesondere die Technologien von Java und CORBA ein enormes Potential für die Erschließung weiterer Plattformen im Rahmen ähnlicher Projekte. Insbesondere könnten innerhalb der verwendeten Datenbank Informationen zu unterschiedlichen Systemen gespeichert und für Analysezwecke zusammengeführt werden. Auf diese Weise wäre es den Systemverantwortlichen einer heterogenen IT-Umgebung möglich, die Integration von Managementdaten teilweise „in Eigenregie“ vorzunehmen, um

- sich einerseits eine erhöhte Flexibilität bei der Auswahl der eingesetzten Werkzeuge zu bewahren und
- andererseits eine verbesserte Anpassung des NSMs an die konkrete IT-Landschaft zu erzielen.

Aber auch das Modellierungstool bietet Ansatzpunkte zur Erweiterung der Funktionalität. Derzeit werden als Ressourcen nur freigegebene Verzeichnisse untersucht, eine Ausdehnung auf die enthaltenen Unterverzeichnisse und Dateien wäre eine erste Ergänzung.
Im Rahmen der Auswertung und weiteren Verarbeitung der (bisher) ermittelten Informationen wäre die Entwicklung eines Skriptgenerators interessant, der Stapeldateien zur Erledigung verschiedener Administrationsaufgaben erzeugt, beispielsweise um

- Datensätze in Sicherheitsprofilen von TSecMan anzulegen (via CLI)
- eine Rekonstruktion von Verzeichnisberechtigungen zu ermöglichen
- eine Umbenennung von lokalen oder globalen Gruppen unter Berücksichtigung erteilter Zugriffsberechtigungen durchzuführen (dabei: Gruppe mit gewünschter Bezeichnung anlegen, Mitglieder übernehmen, DACLs aller betreffenden Verzeichnisse anpassen)
- Veränderungen an der Gruppenstruktur vorzunehmen, z.B. bestehende Gruppen zu teilen (die Gruppenstruktur feiner zu untergliedern)

Ob ein solches Skript

- etwa durch Verwendung des net-Befehls, Windows NT direkt anspricht,
- auf TSecMan spezialisiert ist oder
- andere Werkzeuge verwendet, die gegebenenfalls zu erstellen sind,

kann dabei variiert werden.

Leider gibt es kein universelles Werkzeug für das Netzwerk- und

Systemmanagement einer heterogenen IT-Umgebung. Auch künftig wird es in diesem Bereich ein Nebeneinander von Technologien wie beispielsweise Verzeichnisdiensten, SNMP, CORBA geben. Neue Produktversionen von Anwendungen und Betriebssystemen bieten möglicherweise auch eine verbesserte Unterstützung für solche Technologien – oder besitzen zumindest eigene Schnittstellen (APIs), die von NSM-Werkzeugen genutzt werden können (bzw. müssen).

Im Rahmen des NSMs heterogener IT-Landschaften liegt der Schwerpunkt darauf, die jeweils *vorhandene* Hard- und Software zu managen. Die dabei erforderliche Integration von Managementinformationen bietet – selbst beim Einsatz leistungsfähiger NSM-Produkte von etablierten Herstellern – noch genügend Raum für die Entwicklung eigener NSM-Werkzeuge, die dabei durchaus auf plattformübergreifenden Technologien beruhen können.

Danksagung

Ich möchte mich an diese Stelle bei allen Personen, die mich während meiner Studienzeit und insbesondere im Erstellungszeitraum dieser Arbeit unterstützt haben, bedanken.

Zu Dank verpflichtet bin ich Herrn Professor Dr.-Ing. W.G. Spruth, sein Fachwissen, sein Enthusiasmus und seine zahlreichen persönlichen Kontakte zu Ansprechpartnern aus Wissenschaft und Wirtschaft haben maßgeblich zur Entstehung dieser Arbeit beigetragen.
Weiterhin möchte ich mich bei Herrn W. Diefenbach (von der R+V Versicherung) bedanken, durch sein Engagement auf dem Gebiet des Systemmanagements und seine Zusammenarbeit mit der Universität Leipzig ist das Thema der vorliegenden Arbeit zustande gekommen.
Wertvolle Anregungen sowie kritische Hinweise erhielt ich von Herrn Professor Dr.-Ing. habil. K. Irmscher, wofür ich mich an dieser Stelle ausdrücklich bedanken möchte.
Einen besonderen Dank möchte ich Herrn Professor Dr. rer. nat. K. Hänßgen für die sehr gute Betreuung der Arbeit aussprechen. Obwohl er vielbeschäftigt war, stand er mir stets hilfreich mit Rat und Tat zur Seite.
Danken möchte ich auch Herrn K. Beschorner und Herrn M. Keck, die mir im Vorfeld dieser Arbeit halfen, erste Erfahrungen bei der Programmierung mit CORBA zu sammeln.

Insbesondere möchte ich mich hier auch für die fortwährende Unterstützung und den Rückhalt bedanken, die ich aus dem Familienkreis und speziell von meinen Eltern erfahren habe.

Abkürzungsverzeichnis

ACE	Access Control Entry
ACL	Access Control List; (Zugriffskontrolliste)
API	Application Programming Interface; (Anwendungsprogrammier-schnittstelle)
AMI	Asynchronous Method Invocation
ASN.1	Abstract Syntax Notation One
BDC	Backup Domain Controller; (Backup-Domänen-Controller)
BER	Basic Encoding Rules
BOA	Basic Object Adapter
CCM	CORBA Component Model
CMIS/CMIP	Common Management Information Services/Common Management Information Protocol
CLI	Command Line Interface
CORBA	Common Object Request Broker Architecture
CPU	central processing unit
DACL	Discretionary Access Control List
DB	Datenbank
DCE	Distributed Computing Environment
DFÜ	Datenfernübertragung
DII	Dynamic Invocation Interface
DNS	Domain Name System
DoD	Department of Defense; (Verteidigungsministerium der USA)
DSI	Dynamic Skeleton Interface
DV	Datenverarbeitung
E/A	Eingabe/Ausgabe
EDV	elektronische Datenverarbeitung
EGP	Exterior Gateway Protocol
EP	Endpoint
ESIOP	Environment Specific Inter-ORB Protocol
FAT	File Allocation Table
GIOP	General Inter-ORB Protocol
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer; (Hardware-Abstraktionsschicht)
HTTP	Hypertext Transfer Protocol
IAB	Internet Architecture Board (bis1992: Internet Activities Board): Organisation zur Koordinierung der Entwicklungen der im Internet genutzten Technologien;

	entstand aus dem “Internet Configuration Control Board“ (ICCB); innerhalb der Organisationsstruktur des IAB gibt es eine Untergliederung in zwei technische Gruppen: Internet Engineering Task Force (IETF) und Internet Research Task Force (IRTF)
IDL	Interface Definition Language
IIOP	Internet Inter-ORB Protocol
INS	Interoperable Naming Service
IOM	Inter-Object Message
IOR	Interoparable Object Reference
IP	Internet Protocol
IR	Interface Repository
ISO	International Organization for Standardization
IT	information technology; (Informationstechnologie)
ITU	International Telecommunication Union
JDBC	Java Database Connectivity (Dies ist keine offizielle Abkürzung.)
JDK	Java Development Kit
JNI	Java Native Interface
LAN	Local Area Network
lcfd	Lightweight Client Framework Daemon
LDAP	Lightweight Directory Access Protocol
LPC	Local Procedure Call
LSA	Local Security Authority; (lokale Sicherheitsautorität)
MAC	Mandatory Access Control
MIB	Management Information Base
MVS	Multiple Virtual Storage
NCSC	National Computer Security Center
NM	Netzwerkmanagement
NMS	Netzwerkmanagementstation
NSM	Netzwerk- und Systemmanagement
NT	New Technology
NTFS	NT File System
ODBC	Open Database Connectivity
OID	Object-Identifier
OMA	Object Management Architecture
OMG	Object Management Group
ORB	Object Request Broker
OSI	Open Systems Interconnection
PDA	Personal Digital Assistant
PDC	Primary Domain Controller; Primärer Domänen-Controller

PDU	Protocol Data Unit
POA	Portable Object Adapter
POSIX	Portable Operating System Interface
QoS	Quality of Service
RDBMS	Relational Database Management System
RFC	Request for Comments
RIM	RDBMS Interface Module
SACL	System-ACL
SAM	Security Accounts Manager; (Sicherheitskonten-Manager)
SD	Sicherheitsdeskriptor; (engl. Security Descriptor)
SID	Sicherheitsidentifikation
SM	Systemmanagement
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
TCP	Transmission Control Protocol
TEC	TME 10 Enterprise Console
TII	Time-Independent Invocation
TMA	Tivoli Management Agent
TME	Tivoli Management Environment
TME 10	(Name einer Produktfamilie unter dem der Hersteller „Tivoli Systems“ Software-Produkte für das Netzwerk- und Systemmanagement anbietet)
TMF	Tivoli Management Framework
TMR	Tivoli Management Region
TNR	Tivoli Name Registry
TSecMan	TME 10 Security Management (Dies ist keine offizielle Abkürzung, sie wird nur in der vorliegenden Arbeit verwendet.)
TUA	TME 10 User Administration (Dies ist keine offizielle Abkürzung, sie wird nur in der vorliegenden Arbeit verwendet.)
UDP	User Datagram Protocol
UNC	Universal Naming Convention
URL	Uniform Rescource Locator
US	United States
VM	Virtual Memory
WAN	Wide Area Network
Win32	32-Bit-Windows
WWW	World Wide Web

Literaturverzeichnis

[Ave98] Peter Averkamp, Arno Puder, Kay Römer, Kersten Auel: „Von Big Blue bis GPL: Object Request Broker“, Verlag Heinz Heise, iX 10/1998

[Bes98] Klaus Beschorner: „Realisierung einer Client/Server-Anwendung mit CORBA und Java unter Berücksichtigung bestehender C++-Komponenten“, Diplomarbeit, Eberhard-Karls-Universität Tübingen, 1998

[Bor99] Achim Born: „Management heterogener IT-Strukturen“, Verlag Heinz Heise, iX 07/1999

[Bub96] Frank Bublys: „Modellierung von Zugriffsberechtigungen“, Diplomarbeit, Institut für Informatik, Universität Leipzig, 1996

[Che96] William R. Cheswick, Steven M. Bellovin: „Firewalls und Sicherheit im Internet“, Addison-Wesley, 1996

[Cyp92] R. J. Cypser, „Communications for Cooperating Systems OSI, SNA, TCP/IP“, Addison-Wesley, 1992, ISBN 0-201-50775-7

[Dap97] Thomas Dapper, Carsten Dietrich, Bert Klöppel u.a.: „Windows NT 4.0 im professionellen Einsatz“, Hanser, 1997

[Fla1] David Flanagan: „Java in a Nutshell“, O’Reilly, 1998, deutsche Ausgabe

[Fla2] David Flanagan: „Java Examples in a Nutshell“, O’Reilly, 1998, deutsche Ausgabe

[Fle98] Thomas Fleischer: „Netzwerkinventarisierung und –dokumentation Design und Implementierung einer plattformunabhängigen Softwarelösung“, Diplomarbeit, Institut für Informatik, Universität Leipzig, 1998.

[Gor98] Rob Gordon: „Essential JNI: Java Native Interface“, Prentice-Hall, 1998

[Ham99] Graham Hamilton, Rick Cattell, Maydene Fisher: „JDBC (TM) Datenbankzugriff mit Java (TM)“, Addison-Wesley, 1998

[Heg99] Heinz-Gerd Hegering, Sebastian Abeck, Bernhard Neumair: „Integriertes Management vernetzter Systeme“, dpunkt-Verlag, 1999

[Jan93] Rainer Janssen, Wolfgang Schott: „SNMP – Konzepte, Verfahren. Plattformen“, DATACOM-Verlag, 1993,

[Ker97] Christoph Kersten: „Oracle-Applikationen für PC-Arbeitsgruppen“, Addison-Wesley, 1997

[Kha02] Salman Khan: „Accessing Oracle from Java“, Oracle Corporation, www.oracle.com, (2002)

[Kra98] Wolfgang Kramer, Georg Lodde: „Dritter Anlauf - Version 3 des Simple Network Management Protocol“, Verlag Heinz Heise, iX 02/1998

[Kup98] Martin Kuppinger: „Directory Services“, Computerwoche Verlag, gateway 11/98

[Kup00] Martin Kuppinger: „Microsoft Windows 2000 Server – Das Handbuch“, Microsoft Press, 2000

[Lex94] Klaus Lipinski (Hrsg.): „Lexikon der Datenkommunikation“, DATACOM, 1994.

[Lex98] Klaus Lipinski (Hrsg.): „Lexikon Lokale Netze“, International Thomson Publishing, 1998

[Lon99] Kevin Loney, „Oracle-8-DBA-Handbuch“ (Version 7 bis Version 8), Hanser, 1999, deutsche Ausgabe

[Lor99] Diana Lorentz : „Oracle8i SQL Reference, Release 3 (8.1.7)“, Oracle Corporation, 2000

[Mäu99] Mäurers u.a.: „Java Das Grundlagen Buch“, DATA BECKER, 1999

[Meg98] Ashley J. Meggitt, Timothey D. Ritchey: „Windows NT Benutzer-Administration“, O’Reilly, 1998, deutsche Ausgabe

[MSD98] „Microsoft Developer Network Library“, Microsoft, 1998

[OMG1] OMG: „The Common Object Request Broker: Architecture and Specification, Editorial Revision 2.4.2“, www.omg.org , 2001

[OMG2] OMG: „A Discussion of the Object Management Architecture“, www.omg.org , 1997

[Orf98] Robert Orfali, Dan Harkey: „Client/Server Programming with Java and CORBA“, John Wiley & Sons, 1998

[Ree97] George Reese: „Database Programming with JDBC and Java“, O’Reilly & Associates, 1997

[Ric95] Jeffrey Richter: „Advanced Windows: the developer’s guide to the Win32 API for Windows NT and Windows 98“, Microsoft Press, 1995

[RFC1052] Vinton Cerf:
„IAB Recommendations for the Development of Internet Network Management Standards“, IAB (Request for Comments 1052), 1988

[RFC1155] Marshall T. Rose, Keith McCloghrie:
„Structure and Identification of Management Information for TCP/IP-based Internets“, IAB (Request for Comments 1155), 1990

[RFC1157] Jeffrey D. Case, Mark Fedor, Martin Lee Schoffstall, James R. Davin: „A Simple Network Management Protocol (SNMP)“, IAB (Request for Comments 1157), 1990

[RFC1213] Marshall T. Rose, Keith McCloghrie: „Management Information Base for Network Management of TCP/IP-based internets: MIB-II“, IAB (Request for Comments 1213), 1991

[RFC2570] Jeffrey D. Case, Russ Mundy, David Partain, Bob Stewart:
„Introduction to Version 3 of the Internet-standard Network Management Framework“, IAB (Request for Comments 2570), 1999

[RFC2574] Uri Blumenthal, Bert Wijnen: „User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)“, IAB (Request for Comments 2574), 1999

[Ros94] Marshall T. Rose: „The Simple Book: An Introduction to Internet Management“, Prentice-Hall, 1994

[Sei99] Uwe Seimet: „Services: Basisdienste für den ORB“, Verlag Heinz Heise, iX 01/1999

[Sei00] Uwe Seimet: „Erweiterter Naming Service in CORBA 2.4“, Verlag Heinz Heise, iX 11/2000

[Sie00] Jon Siegel: „CORBA 3 Fundamentals and Programming“, John Wiley & Sons, 2000

[Sos00] Dr. Dieter Sosna: Skript zur „Vorlesung Datenschutz und Datensicherheit“, Institut für Informatik, Universität Leipzig, 2000

[Sta1] Michael Stal: „CORBA 3, Teil 1: Komponenten“, Verlag Heinz Heise, iX 04/2000

[Sta2] Michael Stal: „CORBA 3, Teil 2: Internet und Quality of Service“, Verlag Heinz Heise, iX 05/2000

[Sta93] William Stallings: „SNMP, SNMPv2, AND CMIP: The Practical Guide to Network-Management Standards“, Addison-Wesley, 1993

[SUN1] Sun Microsystems: „Java(TM) 2 SDK, Standard Edition Documentation Version 1.3.1“, java.sun.com, 2002

[SUN2] Sun Microsystems: „The Java Tutorial“, java.sun.com, 2002

[Tie98] Eric Tierling: „Networking mit Windows NT 4.0 Einrichtung, Verwaltung, Referenz“, Addison-Wesley, 1998

[Tiv1] Richard Hawes u.a.: „Managing Access from Desktop to Datacenter: Introducing TME 10 Security Management“, IBM International Technical Support Organization, 1997

[Tiv2] Bart Jacob u.a.: „TME 10 Framework Version 3.2: An Introduction to the Lightweight Client Framework“, IBM International Technical Support Organization, 1997

[Tiv3] Richard Hawes u.a.: „Tivoli Enterprise Internals and Problem Determination“, IBM International Technical Support Organization, 1999

[Tiv4] Rolf Lendenmann u.a.: „An Introduction to Tivoli’s TME 10“, IBM International Technical Support Organization, 1997

[Tiv5] Stefan Uelpenich u.a.: „An Industry Around the Tivoli Framework: Examples from the 10/Plus Association“, IBM International Technical Support Organization, 1998

[Tiv6] „TME 10 Framework Users Guide Version 3.2“, Tivoli Systems, 1997

[Tiv7] Yoichiro Ishii, Hiroshi Kashima: „All About Tivoli Management Agents“, IBM International Technical Support Organization, 1999

[Tiv8] „TME 10 Framework Planning and Installation Guide Version 3.6“, Tivoli Systems, 1998

[Tiv9] „TME 10 Security Management Users Guide Version 3.2“, Tivoli Systems, 1997

[Tiv10] Richard Hawes u.a.: „Tivoli Security Management Design Guide“, IBM International Technical Support Organization, 1998

[Tiv11] „TME 10 Security Management User’s Guide Version 3.6“, Tivoli Systems, 1998

[Tiv12] Richard Hawes u.a.: „Tivoli User Administration Design Guide“, IBM International Technical Support Organization, 1998

[VBJ1] VisiBroker for Java 4.5 Product Documentation: „Installation Guide“, Borland Software Corporation, www.borland.com, 2000

[VBJ2] VisiBroker for Java 4.5 Product Documentation: „Gatekeeper Guide“, Borland Software Corporation, www.borland.com, 2000

[VBJ3] VisiBroker for Java 4.5 Product Documentation: „Programmer’s Guide“, Borland Software Corporation, www.borland.com, 2000

[VBJ4] VisiBroker for Java 4.5 Product Documentation: „Reference“, Borland Software Corporation, www.borland.com, 2000

[Vog98] Andreas Vogel, Keith Duddy: „JAVA Programmming with CORBA“, John Wiley & Sons, 1998

[Zen97] Andreas Zenk: „Sicherheit unter Windows NT 4.0 Sicherheitsmanagement für Unternehmensnetze“, Addison-Wesley, 1997

Anhang

Anhang A: Kurzübersicht zu Quellen im WWW (Auswahl)

Quelle (Bemerkung)	URL
Borland (VisiBroker)	www.borland.com
IBM International Technical Support Organization	www.redbooks.ibm.com
Object Management Group (CORBA)	www.omg.org
Oracle Corporation	www.oracle.com
RFCs (z.B. beim DENIC)	www.denic.de
Sun Microsystems (Java)	www.sun.com
The Simple Times (Hinweise zu SNMP-relevanten RFCs)	www.simple-times.org
Tivoli Systems	www.tivoli.com
Universität Leipzig	www.uni-leipzig.de

Tabelle 18: Kurzübersicht zu Quellen im WWW (Auswahl)

Anhang B: Dateiattribute

Für Dateien und Verzeichnisse können „Attribute“ (sog. „Dateiattribute“) vergeben werden, um den Dateien bzw. Verzeichnissen bestimmte Eigenschaften zuzuordnen. Die Attribute existieren und gelten unabhängig von den NTFS-Berechtigungen. Sie gelten immer für alle Benutzer und Gruppen und haben Vorrang gegenüber den Zugriffsberechtigungen. Die Attribute eines Verzeichnisses beziehen sich nur auf das Verzeichnis selbst, nicht jedoch auf die enthaltenen Dateien und Unterverzeichnisse [Tie98]. Dateiattribute können sowohl bei Verwendung des Dateisystems „FAT“ als auch bei „NTFS“ vergeben werden.

Dateiattribut (engl. Bezeichnung)	Abkürzung	Bedeutung
Schreibgeschützt (Read Only)	R	Name und Inhalt der Datei können nicht verändert werden.
Archiv (Archive)	A	Dieses Attribut wird (automatisch) gesetzt, wenn der Dateiinhalt geändert wurde. Das wird von Datensicherungsprogrammen genutzt, die das Attribut nach erfolgter Sicherung der Datei wieder zurücksetzen. So ist es möglich nur die Dateien zu sichern, die verändert wurden.
Versteckt (Hidden)	H	Die Datei wird „versteckt“ und erscheint bei der Auflistung von Dateien nicht mehr. Daduch wird das Kopieren oder Löschen der Datei verhindert. Allerdings gibt es sowohl auf der Kommandozeile als auch bei der grafischen Umgebung von Windows NT diverse Möglichkeiten die Datei trotzdem aufzulisten. (z.B. mit dem Kommando „dir /a“)
System (System)	S	Die Datei wird als „Systemdatei“ markiert, beim Kopieren, Löschen und Auflisten der Datei gelten die Ausführungen zu „Versteckt“.
Komprimiert (Compressed)	C	Dieses Attribut existiert nur, wenn NTFS als Dateisystem zum Einsatz kommt. Die Datei wird in komprimierter Form gespeichert.

Tabelle 19: Übersicht zu Dateiattributen ([Zen97], [Tie98])

Anhang C: Die Bits innerhalb der Zugriffsmaske

Dieser Anhang soll eine kurze Übersicht zu Einteilung, Bezeichnung und Bedeutung der Bits innerhalb der Zugriffsmaske (im Hinblick auf NTFS-Berechtigungen) bieten.
Bei der Zugriffsmaske handelt es sich um einen 32-Bit-Wert. Die Bits der Zugriffsmaske werden in 3 Kategorien eingeteilt ([MSD98], [Zen97]):

- spezifische Typen (Bits für spezifische Rechte)
- Standardtypen (Bits für Standardrechte)
- generische Typen (Bits für generische Rechte)

Für spezifischen Typen stehen innerhalb der Zugriffsmaske bis zu 16 Bit zur Verfügung (und zwar vom niederwertigsten Bit 0 bis zum Bit 15), deren Bedeutung vom Objekttyp abhängt. Standardtypen hingegen können auf verschiedene Objekttypen angewendet werden. Generische Typen werden auf Kombinationen aus spezifischen Typen und Standardtypen abgebildet. Die dabei geltende Abbildungsvorschrift ist wiederum vom Objektyp abhängig. Tabelle 20 gibt einen Überblick zu den Standardtypen.

Bezeichnung des Bits	Abkürzung im Modellierungs-tool	Position in der Zugriffs-maske	Bedeutung
SYNCHRONIZE	Sync	20	Dieses Bit wird genutzt, um den Zugriff auf Objekte zu synchronisieren.
WRITE_OWNER	O	19	Recht, den Besitz des Objektes zu übernehmen
WRITE_DACL	P	18	Recht zur Modifikation der DACL des Objektes
READ_CONTROL	RC	17	Recht zum Lesen der Bestandteile des Sicherheitsdeskriptors (mit Ausnahme der SACL)
DELETE	D	16	Recht zum Löschen des Objektes

Tabelle 20: Standardtypen, Bits für Standardrechte ([MSD98], [Zen97])

Für das Bit „READ_CONTROL“ existieren die folgenden alternativen Bezeichnungen:

- STANDARD_RIGHTS_READ
- STANDARD_RIGHTS_WRITE
- STANDARD_RIGHTS_EXECUTE

Für spezifische Typen (Bits für spezifische Rechte) sind – in Bezug auf Datei- und Verzeichnisobjekte – die in Tabelle 21 angegebenen Bezeichnungen definiert.

Bezeichnung des Bits (alternative Bezeichnung bei Verzeichnisobjekten)	Abkürzung im Modellierungstool	Position in der Zugriffsmaske
FILE_WRITE_ATTRIBUTES	FWA	8
FILE_READ_ATTRIBUTES	FRA	7
FILE_DELETE_CHILD	FDC	6
FILE_EXECUTE (FILE_TRAVERSE)	FX	5
FILE_WRITE_EA	FW_EA	4
FILE_READ_EA	FR_EA	3
FILE_APPEND_DATA (FILE_ADD_SUBDIRECTORY)	FA_D	2
FILE_WRITE_DATA (FILE_ADD_FILE)	FW	1
FILE_READ_DATA (FILE_LIST_DIRECTORY)	FR	0

Tabelle 21: Übersicht zu spezifischen Typen, Bits für spezifische Rechte ([MSD98], [Zen97])

Die in Tabelle 21 angegebenen Bits für spezifische Rechte sind – mit Ausnahme von Bit 6 – sowohl auf Datei- als auch auf Verzeichnisobjekte anwendbar. Bit 6 bezieht sich nur auf Verzeichnisse. Wie aus Tabelle 21 hervorgeht, wird beispielsweise Bit 5 vom Modellierungstool immer als „FX“ abgekürzt, d.h. auf die alternativen Bezeichnungen wird verzichtet.
Für generische Rechte stehen insgesamt 4 Bit zur Verfügung, siehe Tabelle 22.

Bezeichnung des Bits	Abkürzung im Modellierungstool	Position in der Zugriffsmaske	Berechtigung
GENERIC_READ	GR	31	Lesen
GENERIC_WRITE	GW	30	Schreiben
GENERIC_EXECUTE	GX	29	Ausführen
GENERIC_ALL	GA	28	Lesen, Schreiben, Ausführen

Tabelle 22: generische Typen, Bits für generische Rechte ([MSD98], [Zen97])

Für Dateiobjekte werden die Bits für generische Rechte wie folgt abgebildet ([MSD98], [Zen97]):

Generic_Read (wird zusammengesetzt aus):
- SYNCHRONIZE
- STANDARD_RIGHTS_READ (also „READ_CONTROL“)
- FILE_READ_DATA
- FILE_READ_ATTRIBUTE
- FILE_READ_EA

Generic_Write (wird zusammengesetzt aus):
- SYNCHRONIZE
- STANDARD_RIGHTS_WRITE
- FILE_WRITE_DATA
- FILE_WRITE_ATTRIBUTE
- FILE_WRITE_EA
- FILE_APPEND_DATA

Generic_Execute (wird zusammengesetzt aus):
- SYNCHRONIZE
- STANDARD_RIGHTS_EXECUTE
- FILE_READ_ATTRIBUTE
- FILE_EXECUTE

Anhang D: Bezug zu Windows 2000

Das Betriebssystem „Windows 2000“ wurde in dieser Arbeit nicht untersucht. Dieser Anhang ist daher auch als Anregung für entsprechende Untersuchungen/Arbeiten zu verstehen.

Die folgenden Zitate stammen aus [Kup00]. Dort wird im Zusammenhang der „Migration von Windows NT-Domänencontrollern“ u.a. auch zum Thema der Zugriffsberechtigungen bezogen auf die beiden Betriebssysteme (Windows NT, Windows 2000) Stellung genommen [Kup00, S.199]:

„Das Konzept der SIDs wird auch bei Windows 2000 weiter verfolgt, wobei auch hier die Domäne Grenze für SIDs ist. Unterschiedliche Domänen bei Windows 2000 verfügen also auch über unterschiedliche SIDs.“

Bezogen auf einen (von Windows NT nach Windows 2000) zu migrierenden Domänencontroller heißt es in [Kup00, S.199]:

„Da sich die SIDs nicht verändern und die Zugriffsberechtigungen auf Ressourcen über SIDs gesteuert werden, ergeben sich folgerichtig auch keine Auswirkungen auf den Ressourcenzugriff. Bei jeder Ressource finden sich ACLs (Access Control Lists, Zugriffskontrolllisten) mit ACEs (Access Control Entries).“

Zusammenfassend wird in [Kup00, S.199] formuliert:

„Innerhalb einer Domäne verändern sich diese Strukturen also prinzipiell überhaupt nicht.“

Anhang E: Inhalt der beiliegenden CD

Die beiliegende CD enthält das vorliegende Dokument im Postscript-Format.
(Das Modellierungstool ist beim Autor verfügbar.)

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Leipzig, Oktober 2002 .....